NON-ARCHIMEDEAN ANALYSIS, T-FUNCTIONS, AND CRYPTOGRAPHY

VLADIMIR ANASHIN

Abstract. These are lecture notes of a 20-hour course at the International Summer School *Mathematical Methods and Technologies in Computer Security* at Lomonosov Moscow State University, July 9–23, 2006.

Loosely speaking, a T-function is a map of n-bit words into n-bit words such that each i-th bit of image depends only on low-order bits $0, \dots, i$ of the pre-image. For example, all arithmetic operations (addition, multiplication) are T-functions, all bitwise logical operations (XOR, AND, etc.) are T-functions. Any composition of T-functions is a T-function as well. Thus T-functions are natural computer word-oriented functions.

It turns out that T-functions are continuous (and often differentiable!) functions with respect to the so-called 2-adic distance. This observation gives a powerful tool to apply 2-adic analysis to construct wide classes of T-functions with provable cryptographic properties (long period, balance, uniform distribution, high linear complexity, etc.); these functions currently are being used in new generation of fast stream ciphers. We consider these ciphers as specific automata that could be associated to dynamical systems on the space of 2-adic integers. From this view the lectures could be considered as a course in cryptographic applications of the non-Archimedean dynamics; the latter has recently attracted significant attention in connection with applications to physics, biology and cognitive sciences.

During the course listeners study non-Archimedean machinery and its applications to stream cipher design.

Contents

Vladimir Anashin is a Professor and Dean of the Faculty of Information Security at the Russian State University for the Humanities.

1. Introduction

1.1. **Goals.** Imagine we are a team of cryptographers, and we are going to design a software-oriented cipher. That is, we are going to combine basic microchip instructions to make a very specific transformation of machine words. On the one hand, this transformation must be fast; that is, the corresponding computer program must achieve high performance. On the other hand, this transformation must be secure: Having both an output (that is, encrypted text) *and the program*, it must be infeasible to obtain *illegally* the corresponding input (i.e., plain text).

At this point, we should understand the following issues:

- What are these basic instructions? What are reasonable compositions of these instructions?
- Could we give an evidence that certain transformation of this kind is secure?

Actually, a goal of the course is to clarify these issues. Moreover, in order to make our considerations not too general, and to conclude with some practical applications, we restrict ourselves with a certain specific kind of ciphers, the so-called *stream ciphers*.

1.2. **What are stream ciphers?** In contemporary digital computers information is represented in a binary form, as a sequence of zeros and ones. So a plaintext is a sequence $\alpha_0, \alpha_1, \alpha_2, \ldots$, where $\alpha_j \in \mathbb{B} = \{0, 1\}$. Let $\Gamma = \gamma_0, \gamma_1, \gamma_2, \ldots$ be another sequence of zeros and ones, which is known both to Alice and Bob, and which is known to no third party. The sequence Γ is called a *keystream*. To encrypt a plaintext, Alice just XORes it with the key:

$$
\begin{array}{lll}
 & \alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_i, \ldots & \text{(plaintext)} \\
\bigoplus & & \text{(bitwise addition modulo 2)} \\
 & \gamma_0, \gamma_1, \gamma_2, \ldots, \gamma_i, \ldots & \text{(keystream)} \\
\hline
 & \zeta_0, \zeta_1, \zeta_2, \ldots, \zeta_i, \ldots & \text{(encrypted text)}
\end{array}
$$

To decrypt, Bob acts in the opposite order:

$$
\begin{array}{lll}
 & \zeta_0, \zeta_1, \zeta_2, \ldots, \zeta_i, \ldots & \text{(encrypted text)} \\
\bigoplus & & \text{(bitwise addition modulo 2)} \\
 & \gamma_0, \gamma_1, \gamma_2, \ldots, \gamma_i, \ldots & \text{(keystream)} \\
\hline
 & \alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_i, \ldots & \text{(plaintext)}
\end{array}
$$

Loosely speaking, Shannon's Theorem states that this encryption is secure providing the keystream Γ is picked at random for each plaintext. In real life settings we very rarely could fulfil conditions of Shannon's Theorem, and usually we use a *pseudorandom* keystream Γ rather than a random one. That is, usually in real life ciphers Γ is produced by a certain algorithm, and Γ only looks like random (e.g., passes certain statistical tests). A *pseudorandom generator*, or a *pseudorandom number generator* (PRNG) is an algorithm that takes a short random string (which is called *a key*, or *a seed*) and stretches it into a much longer sequence, a *keystream*. Actually, within the scope of the course we speak about *stream cipher* meaning the latter is

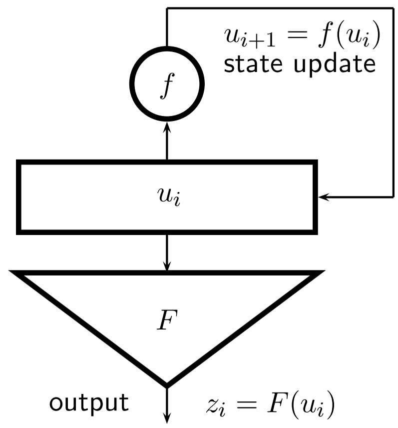

FIGURE 1. Ordinary PRNG

a pseudorandom generator which is used for encryption according to the protocol described above.

Not every PRNG is suitable for stream encryption. Stream ciphers are *cryptographically secure* PRNG's; that is, they must not only produce statisticly good sequences, but also they must withstand *cryptoanalyst's attacks*.

2. Preliminaries

Now we will try to state some of the above mentioned notions more formally. We start with our main notion, a PRNG.

2.1. **Pseudorandom generators.** Basically, a generator we consider during the course is a finite automaton $\mathfrak{A} = \langle N, M, f, F, u_0 \rangle$ with a finite state set N, state transition (or, state update) function $f : N \to N$, finite output alphabet M, output function $F : N \to M$ and an initial state (seed) $u_0 \in N$. Thus, this generator (see Figure 1) produces a sequence

$$\mathcal{S} = \{F(u_0), F(f(u_0)), F(f^{(2)}(u_0)), \ldots, F(f^{(j)}(u_0)), \ldots\}$$

over the set M, where

$$f^{(j)}(u_0) = \underbrace{f(\ldots f(}_{j \text{ times}} u_0) \ldots) \quad (j = 1, 2, \ldots); \quad f^{(0)}(u_0) = u_0.$$

Automata of the form $\mathfrak{A}$ could be used either as pseudorandom generators per se, or as components of more complicated pseudorandom generators, the so called *counter-dependent generators* (see Figure 2); the latter produce sequences $\{z_0, z_1, z_2, \ldots\}$ over M according to the rule

$$z_0 = F_0(u_0), u_1 = f_0(u_0); \ldots z_i = F_i(u_i), u_{i+1} = f_i(u_i); \ldots \tag{2.0.1}$$

That is, at the $(i+1)^{\text{th}}$ step the automaton $\mathfrak{A}_i = \langle N, M, f_i, F_i, u_i \rangle$ is applied to the state $u_i \in N$, producing a new state $u_{i+1} = f_i(u_i) \in N$, and outputting a symbol $z_i = F_i(u_i) \in M$.

Now to make our considerations more practical, we must impose certain restrictions on these state update and output functions. As we want our

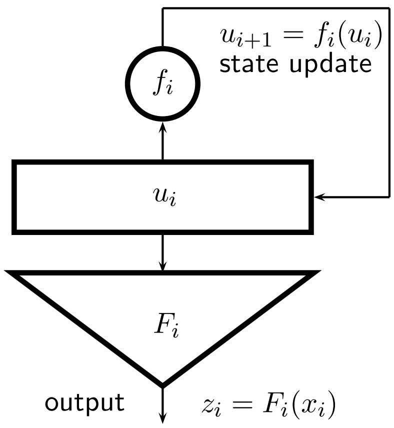

FIGURE 2. Counter-dependent PRNG

generators to be implemented in software and to demonstrate good performance, these functions can not be arbitrary, they must be finally written as more or less short programs. That is, these functions must be represented as (not too complicated) compositions of basic instructions of a contemporary processor. Then, what are these basic instructions?

2.2. **Basic instructions.** A contemporary processor is word-oriented. That is, it works with words of zeroes and ones of a certain fixed length n (usually $n = 8, 16, 32, 64$). Each binary word $z \in \mathbb{B}^n$ of length n could be considered as a base-2 expansion of a number $z \in \{0, 1, \dots, 2^n - 1\}$ and vise versa:

$$z = \zeta_0 + \zeta_1 \cdot 2 + \zeta_2 \cdot 2^2 + \cdots \longleftrightarrow \zeta_0\zeta_1\zeta_2 \ldots \in \mathbb{B}^n$$

We also can identify the set $\{0, 1, \dots, 2^n-1\}$ with residues modulo 2^n; that is with the elements of the residue ring $\mathbb{Z}/2^n\mathbb{Z}$ modulo 2^n. Actually, *arithmetic* (numerical) instructions of a processor are just *operations of the residue ring* $\mathbb{Z}/2^n\mathbb{Z}$: An n-it word processor performing a single instruction of addition (or multiplication) of two n-bit numbers just deletes more significant digits of a sum (or of a product) of these numbers thus merely reducing the result modulo 2^n. Note that to calculate a sum of two integers (i.e., without reducing the result modulo 2^n) a 'standard' processor uses not a single instruction but a program tt consists of basic instructions!

Other sort of basic instructions of a processor are *bitwise logical* operations: XOR, OR, AND, NOT, which are clear from their definitions. It worth notice only that the set $\mathbb{B}^n$ with respect to XOR could be considered also as an n-dimensional vector space over a field $\mathbb{Z}/2\mathbb{Z} = \mathbb{B}$.

The third type of instructions could be called a *machine* ones, since they depend on a processor. But usually they include such standard instructions as shifts (left and right) and circular rotations of an n-bit word.

Some more formal sample definitions: Let

$$z = \delta_0(z) + \delta_1(z) \cdot 2 + \delta_2(z) \cdot 2^2 + \delta_3(z) \cdot 2^3 + \cdots$$

be a base-2 expansion for $z \in \mathbb{N}_0 = \{0, 1, 2, \dots\}$.Then, according to the respective definitions, we have

- $y \,\mathsf{XOR}\, z = y \oplus z$ is a bitwise addition modulo 2: $\delta_j(y \,\mathsf{XOR}\, z) \equiv \delta_j(y) + \delta_j(z) \pmod 2$;
- $y \,\mathsf{AND}\, z$ is a bitwise multiplication modulo 2: $\delta_j(y \,\mathsf{AND}\, z) \equiv \delta_j(y) \cdot \delta_j(z) \pmod 2$;
- $\lfloor \frac{z}{2} \rfloor$, the integral part of $\frac{z}{2}$, is a shift towards less significant bits;
- $2 \cdot z$ is a shift towards more significant bits;
- $y \,\mathsf{AND}\, z$ is masking of z with the mask y;
- $z \pmod{2^k} = z \,\mathsf{AND}(2^k - 1)$ is a reduction of z modulo 2^k

Let us make the first important observation:

> *Basic instructions of a processor, with the exception of rotations, are well defined on the whole set of positive integers.*

Now we look at the basic instructions from a bit another point.

2.3. ***T*-functions.** From a school textbook algorithm of addition of base-2 expansions of positive integers it immediately follows that each i-th bit of the sum does not depend on higher order bits of summands, i.e., on j-th bits with $j > i$. The same holds for products, bitwise logical operations, and shifts towards higher order bits. This observation gives rise to the following definition:

Definition 2.1 (T-function)**.** An (m-variate) T-function is any mapping

$$F\colon (\ldots, \alpha_2^{\downarrow}, \alpha_1^{\downarrow}, \alpha_0^{\downarrow}) \mapsto (\ldots, \Phi_2(\alpha_0^{\downarrow}, \alpha_1^{\downarrow}, \alpha_2^{\downarrow}), \Phi_1(\alpha_0^{\downarrow}, \alpha_1^{\downarrow}), \Phi_0(\alpha_0^{\downarrow}))$$

where $\alpha_i^{\downarrow} \in \mathbb{B}^m$ is a Boolean columnar m-dimensional vector; $\mathbb{B} = \{0, 1\}$; $\Phi_i\colon (\mathbb{B}^m)^{(i+1)} \to \mathbb{B}^n$ maps $(i+1)$ Boolean columnar m-dimensional vectors $\alpha_i^{\downarrow}, \ldots, \alpha_0^{\downarrow}$ to n-dimensional Boolean vector $\Phi_i(\alpha_0^{\downarrow}, \ldots, \alpha_i^{\downarrow})$.

For instance, a univariate T-function $F\colon \mathbb{B}^n \to \mathbb{B}^n$ is a mapping of $\mathbb{B}^n$ into itself such that

$$(\ldots, \chi_2, \chi_1, \chi_0) \stackrel{F}{\mapsto} (\ldots; \psi_2(\chi_0, \chi_1, \chi_2); \psi_1(\chi_0, \chi_1); \psi_0(\chi_0)),$$

where $\chi_j \in \{0, 1\}$, and each $\psi_j(\chi_0, \ldots, \chi_j)$ is a Boolean function in Boolean variables $\chi_0, \ldots, \chi_j$.

Thus, we state that

> *Basic instructions of a processor, with the exception of rotations and shifts towards low order bits, are T-functions.*

Obviously, a composition of T-functions is a T-function; so while combining basic instructions into a program, we very often can say that the resulting mapping (that is, a program) is a T-function. So, it seems to be a good idea to study the above mentioned automata under a restriction that both their state update and output functions are T-functions, and try to design a stream cipher on their base.

Few words about terminology: Despite the term 'T-function' was suggested only in 2002 by A. Klimov and A. Shamir, see [15], these mappings are well-known mathematical objects dating back to 1960$^{\text{th}}$ (however, under other names: Compatible mappings in algebra, determined functions in automata theory, triangle boolean mappings in the theory of Boolean functions, functions that satisfy Lipschitz condition with constant 1 in p-adic analysis; see e.g. [19], [25], [4]). Throughout the course we use the term

'T-function' as the most accepted by cryptographic community; however, we will be interested in those properties of T-functions that are explored in other areas of mathematics. The mentioned p-adic analysis appears to be the most important one.

2.4. **Preparations to p-adic Calculus.** We can calculate a sum of two positive integers represented by their base-2 expansions with a 'school textbook' algorithm. Note that the summands are represented as finite strings of 0's and 1's (or, better to say, as *infinite* strings of 0's and 1's that contain only *finite* number of 1's). Let us look what happens if we apply this algorithm to *arbitrary infinite* strings of 0's and 1's.

Consider an example:

$$\begin{array}{cccccc} & \ldots 1 & 1 & 1 & 1 \\ + & & & & \\ & \ldots 0 & 0 & 0 & 1 \\ \hline & \ldots 0 & 0 & 0 & 0 \end{array}$$

Obviously, the string $\ldots 000$ is merely 0, and the string $\ldots 001$ is 1. But then we *must* conclude that $\ldots 111 = -1$; that is, the infinite string $\ldots 111$ is a base-2 expansion of a *negative* integer -1. With this in mind, we continue our investigations. Let's try multiplication now:

$$\begin{array}{ccccccc} & \ldots 0 & 1 & 0 & 1 & 0 & 1 \\ \times & & & & & & \\ & \ldots 0 & 0 & 0 & 0 & 1 & 1 \\ \hline & \ldots 0 & 1 & 0 & 1 & 0 & 1 \\ + & & & & & & \\ & \ldots 1 & 0 & 1 & 0 & 1 & \\ \hline & \ldots 1 & 1 & 1 & 1 & 1 & 1 \end{array}$$

As we know that $\ldots 0011 = 3$, and, as we have agreed, $\ldots 111 = -1$, then we are forced to conclude that $\ldots 01010101 = -\frac{1}{3}$. This sounds somewhat odd for us, but *not so* for a computer! These calculations could be made with an ordinary Windows built-in calculator, up to the best precision it admits, 64 bits.[1]

Now denote $\mathbb{Z}_2$ the set of all infinite binary strings. We could define addition and multiplication on $\mathbb{Z}_2$ with the said school-textbook algorithms, thus turning $\mathbb{Z}_2$ into a ring. Obviously, any T-function is well defined on $\mathbb{Z}_2$. Summing it up, we conclude that

[1] Don't forget to switch the calculator into `scientific mode` and choose `bin`.

Basic processor instructions, with the only exception of rotations, as well as T-functions, are well defined functions on the set $\mathbb{Z}_2$ of all infinite binary sequences; these functions are evaluated in $\mathbb{Z}_2$.

As a matter of fact, these functions turn out to be *continuous* in some well-defined sense. Moreover, very often they are *differentiable* functions, and we can use a special sort of Calculus to study their properties that are crucial for cryptography with the techniques similar to that of classical Calculus. That is what we are going to do within the course.

What we are thinking about when saying 'Calculus'? Well, of derivations, for instance. And what notion do we use in the definition of a derivative? Evidently, a notion of limit. But saying that 'a is a limit of the sequence $\{a_i\}_{i=0}^{\infty}$ of numbers as i goes to infinity' we just mean that these a_i are *approximations* of a, and we can achieve an *arbitrarily* good precision of these approximations by taking *sufficiently large* i.

Now we are going to understand what does this 'precision' means, or, better to say, what a computer thinks of what 'precision' means. A computer can not work with arbitrarily long binary words. Actually, its basic instructions work with words of certain length, a *bitlength*. Usual values of bitlengths of contemporary processors are 8,16,32, 64.

Now take some binary string, e.g., a string $\underbrace{1\dots111}_{64\text{ times}}$; that is, a number $2^{64}-1 = 18446744073709551615$. A 8-bit processor can work only with 8-bit string, so it can store only 8 less significant bits of this string; that is, the number $2^8 - 1 = 255$. A 16-bit processor stores 16 bit, that is, the number $2^{16}-1 = 65535$; a 32-bit processor stores this string as $2^{32}-1 = 4294967295$, etc. It is reasonable to say that 255 is an approximation with 8-bit precision of the number $2^{64} - 1$, 65535 is an approximation with 16-bit precision, etc.

Following this logic, we finally conclude that the sequence

$$255, 65535, 4294967295, \dots, 2^{2^n} - 1, \dots$$

tends to $-1 = \dots 111$ as k goes to infinity, and the same does the sequence $2^n - 1$. That is, $\overset{2}{\lim\limits_{n\to\infty}}(2^n - 1) = -1$, where $\overset{2}{\lim}$ is something that behaves like an ordinary limit, but with respect to the 'n-bit precision'. Further, in case we want this $\overset{2}{\lim}$ behave similarly to an ordinary limit, we must conclude that $\overset{2}{\lim\limits_{n\to\infty}} 2^n = 0$, which is extremely odd![2]

To discover the underlying reality, we now must understand on what notion is the notion of limit based. Recalling the classical definition, we see that the notion of limit is stated in terms of 'how close the two numbers are'. That is, the notion of limit is based on the notion of distance!

The above examples demonstrate that for human beings and for computers, 'distance' means quite different things, or, better to say, is measured in different ways. For us, human beings, a number $2^{32} = 4294967296$ lies at a bigger distance from 0 than the number $2^8 = 256$; on the contrary, 2^{32} is

[2]Not too odd, however. Intuitively, the sequence $\dots0001, \dots0010, \dots0100, \dots$, which is the sequence of base-2 expansions of $1, 2, 4, 8, \dots$, tends to $\dots0000 = 0$!

closer to 0 than 2^8, for a computer. What a peculiar distance a computer uses?

3. The notion of p-adic integer

3.1. **The notion of distance.** Actually, when we measure a distance between two points, we associate a non-negative real number to the pair of points. Obviously, this number is 0 if and only if these points coincide, and the distance measured from the first point to the second one is equal to the distance measured in the opposite direction, from the second point towards the first. The distance obeys the 'law of a triangle'; that is, the distance from the first point A to the second point B is not greater than the sum of two distances, from the first point A to an arbitrary third point C, and from this third point C to the point B. These observations are summarized in the following definition[3]:

Definition 3.1 (Metric)**.** Let M be a non-empty set, and let $d\colon M\times M\to\mathbb{R}_{\geq 0}$ be a function valuated in non-negative real numbers. The function d is called a *metric* (and M is called a *metric space*) whenever d obeys the following laws:

(1) For every pair $a,b\in M$, $d(a,b)=0$ if and only if $a=b$.
(2) For every pair $a,b\in M$, $d(a,b)=d(b,a)$.
(3) For every triple $a,b,c\in M$, $d(a,b)\leq d(a,c)+d(c,b)$.

For example, the set $\mathbb{R}$ of all real numbers is a metric space with metric $d(a,b)=|a-b|$, where $|\cdot|$ is absolute value. The latter notion also could be defined for arbitrary commutative ring R.

Definition 3.2 (Norm)**.** A function $\|\cdot\|$ defined on the R and valuated in $\mathbb{R}_{\geq 0}$ is called a *norm* whenever $\|\cdot\|$ satisfies the following conditions:

(1) For every $a\in R$, $\|a\|=0$ if and only if $a=0$.
(2) For every pair $a,b\in R$, $\|a\cdot b\|=\|a\|\cdot\|b\|$.
(3) For every pair $a,b\in R$, $\|a+b\|\leq\|a\|+\|b\|$.

It is easy to verify that assuming $d(a,b)=\|a-b\|$ we define metric d on the ring R. This metric d is called a metric induced by the norm $\|\cdot\|$.

Note that once the norm (whence, metric) on the ring R is defined, we immediately define a notion of convergent sequence over R, a notion of limit, a notion of continuous function defined on R and valuated in R, a notion of derivative of a function, etc. For instance, element $a\in R$ is a derivative of the function $f\colon R\to R$ at the point $x\in R$ if and only if for all *sufficiently small* $h\in R$, $h\neq 0$, (that is, for $\|h\|<\delta$ for some real $\delta>0$)

$$f(x+h)=f(x)+a\cdot h+\lambda(h),$$

where $\frac{\|\lambda(h)\|}{\|h\|}$ goes to 0 as $\|h\|$ goes to 0. Thus, loosely speaking, every new norm leads to a new Calculus.

[3]Mathematicians used to speak of metric rather than of distance, but distance is also OK

3.2. **Norms on $\mathbb{Z}$.** We know that absolute value $|\cdot|$ is a norm on the ring $\mathbb{Z}$ of all integers. The question arises, is $|\cdot|$ *the only* norm on $\mathbb{Z}$? Surprisingly, *not*!

Let p be a prime number. Using this p, we define now a norm $\|\cdot\|_p$ on $\mathbb{Z}$. Obviously, since $\|-a\| = \|a\|$ for every $a \in R$ (this is an exercise to deduce the identity from Definition 3.2!), it suffices to define $\|\cdot\|_p$ on the set $\mathbb{N}_0$ of all non-negative integers. We assume $\|0\|_p = 0$. Now, if $n > 0$ is a natural number, it has a unique representation as a product of powers of pairwise distinct primes. Denote $\operatorname{ord}_p n$ exponent of p in this representation and put $\|n\|_p = p^{-\operatorname{ord}_p n}$. It is an exercise to verify that the so defined function is a norm.

Indeed, (1) and (2) of Definition 3.2 obviously hold for the so defined norm. Moreover, (3) holds in a *stronger* form:

(3′) For every pair $a, b \in \mathbb{Z}$, $\|a + b\|_p \le \max\{\|a\|_p, \|b\|_p\}$.

From here it obviously follows that the metric d_p defined by the norm $\|\cdot\|_p$ also satisfies a stronger relation than (3) of Definition 3.1:

(3′) For every triple $a, b, c \in \mathbb{Z}$, $d_p(a, b) \le \max\{d_p(a, c), d_p(c, b)\}$.

The latter relation is called a *strong triangle inequality*, and a metric that satisfies this inequality is called a *non-Archimedean* metric, or an *ultrametric*. Accordingly, a metric space equipped with a non-Archimedean metric is called a non-Archimedean metric space, or an ultrametric space.

3.3. **p-adic integers.** Clearly, for natural $n \in \mathbb{N}$ one can calculate $\operatorname{ord}_p n$ according to the following rule: Represent n in its base-p expansion, find the least significant non-zero digit (let it be the i-th digit; enumeration starts with zero); then $\operatorname{ord}_p n = i$. That is,

$$n = \ldots a_{i+1} a_i \underbrace{0 \ldots 0}_{i \text{ zeros}}; a_i \neq 0 \Rightarrow \|n\|_p = \frac{1}{p^i}.$$

The latter definition could be expanded on the whole set $\mathbb{Z}_p$ of *infinite* strings of digits $0, 1, \ldots, p-1$ in an obvious manner. Now it is not difficult to prove that the set $\mathbb{Z}_p$ is a commutative ring with respect to addition and multiplication defined by 'school-textbook' algorithms, and, moreover, the so defined function $\|\cdot\|_p$ is a norm on this ring![4] Elements of the ring $\mathbb{Z}_p$ are called *p-adic integers*. Actually, we think of the infinite string $\ldots a_i a_{i-1} \ldots a_0$ over an alphabet $\{0, 1, \ldots, p-1\}$ as of base-p expansion of a p-adic integer a:

$$a = \ldots a_i a_{i-1} \ldots a_0 = \sum_{i=0}^{\infty} a_i \cdot p^i \tag{3.2.1}$$

Note that for $a, b \in \mathbb{Z}_p$ $d_p(a, b) = \frac{1}{p^i}$ for some $i = 0, 1, 2, \ldots, \infty$ (case $i = \infty$ just means that $d_p(a, b) = 0$, whence, $a = b$). Moreover, $d_p(a, b) = \frac{1}{p^i}$ if and only if

$$a = \ldots a_{i+1} a_i c_{i-1} \ldots c_0;$$
$$b = \ldots b_{i+1} b_i c_{i-1} \ldots c_0,$$

[4] Prove this.

and $a_i \neq b_i$. Using an obvious analogy with non-negative *rational* integers we write in this case that $a \equiv b \pmod{p^i}$. Thus, $d_p(a,b) = \frac{1}{p^i}$ where i is the biggest non-negative rational integer such that $a \equiv b \pmod{p^i}$, and $a \not\equiv b \pmod{p^{i+1}}$. Throughout the course we denote the i-th digit ($i = 0, 1, 2, \ldots$) in a base-p expansion of a p-adic integer $a \in \mathbb{Z}_p$ via $\delta_i^p(a)$; that is, $\delta_i^p(a) = a_i$, cf. (3.2.1). We omit the superscript (especially in case $p = 2$) when it does not lead to misunderstandings.

The ring $\mathbb{Z}_2$ of infinite binary strings mentioned above corresponds to the case $p = 2$. Thus, $\mathbb{Z}_2$ is an ultrametric space with respect to the metric d_2 defined by the norm $\|\cdot\|_2$. And, indeed, with respect to this metric d_2 the sequence $1, 2, 4, \ldots, 2^n, \ldots$ converges to 0 as n goes to infinity; whence[5], the sequence $1, 3, 7, \ldots, 2^n - 1, \ldots$ indeed converges to -1.

Actually *a processor works with approximations of 2-adic integers with respect to 2-adic metric*: When one tries to load a number which base-2 expansion contains more than n significant bits into a registry of an n-processor, the processor just writes only n low order bits of the number in a registry *thus reducing the number modulo* 2^n. Thus, precision of the approximation is defined by the bitlength of the processor.

Since the ring (metric space) $\mathbb{Z}_2$ is of most importance for us, we proceed with some examples that illustrate our main notions with respect to $\mathbb{Z}_2$.

Sequences that contain only finite number of 1's correspond to non-negative rational integers represented by their base-2 expansions:

$$\ldots 00011 = 3$$

Sequences that contain only finite number of 0's correspond to negative rational integers[6]:

$$\ldots 111100 = -4$$

Sequences that are (eventually) periodic correspond to rational numbers that could be represented by irreducible fractions with odd denominators[7]:

$$\ldots 1010101 = -\frac{1}{3}$$

Non-periodic sequence correspond to no rational number.

An example one how we measure distances in $\mathbb{Z}_2$:

$$\left.\begin{array}{l} \ldots 10101\underbrace{0101} = -\dfrac{1}{3} \\ \ldots 00000\underbrace{0101} = 5 \end{array}\right\} \Longrightarrow d_2\left(-\frac{1}{3}, 5\right) = \frac{1}{2^4} = \frac{1}{16}$$

That is, $-\frac{1}{3} \equiv 5 \pmod{16}$; $-\frac{1}{3} \not\equiv 5 \pmod{32}$.

3.4. **Odd world.** Finally we conclude that our computers live in the world other than we human beings. This virtual world is very odd. In this subsection we only mention some facts about this virtual world to make it more familiar to us. Proofs (and other peculiar facts) could be found in the above mentioned books and monographs on p-adic analysis.

[5]To prove this we must prove a theorem on limit of sum of two convergent sequences before. It is a good exercise to re-prove all classical theorems about limits of compositions of sequences in general case, for arbitrary metric!

[6]Prove this

[7]Prove this

Our world, the world of real numbers $\mathbb{R}$ is Archimedean. That is, it satisfies the Archimedean Axiom which read:

> Given a segment S of real line of length s, and another (smaller) segment L of length ℓ, $\ell < s$, there exists a natural number n such that $n \cdot \ell > s$. (That is, if we append a short segment to itself sufficient number of times, we can make the resulting segment arbitrarily long).

This axiom *does not hold* in the p-adic world $\mathbb{Z}_p$: Appending a segment to itself we could make the resulting segment *shorter* than the original one! For instance, let $p = 2$ and let L be some 'segment of length $\frac{1}{2}$', say, $L = 2$ then doubling the segment ('appending' it to itself) we, obtain a 'segment' $2 \cdot L = 4$, and for which we have $\|4\|_2 = \frac{1}{2}$. The 'doubled segment' is twice as *short* as the original!

Of course, origin of this fact is hidden in a strong triangle inequality (3′) that governs the non-Archimedean world. This inequality implies other odd-looking facts, e.g.,

- All triangles are isosceles!
- Every point inside a ball is a center of this ball!
- The series $\sum_{i=0}^{\infty} z_i$ of p-adic integers are convergent *if and only if* $\overset{p}{\lim\limits_{i\to\infty}} z_i = 0$ (where $\overset{p}{\lim\limits_{i\to\infty}}$) is a limit with respect to the p-adic norm $\|\cdot\|_p$).

By the way, this implies that, say, $\ln(-3) = -\sum_{i=1}^{\infty} \frac{4^i}{i}$ is a 2-adic integer!

If you are going to prove these statements (which is a good exercise!) note that every ball of radius $\frac{1}{p^k}$ in $\mathbb{Z}_p$ is of the form $a+p^k\cdot\mathbb{Z}_p = \{a+p^k\cdot z\colon z \in \mathbb{Z}_p\}$. By the way, from here it follows that, in case $p = 2$, a boundary of a (closed) ball is itself a ball of radius $\frac{1}{p^{k+1}}$; e.g., a sphere of radius $\frac{1}{2}$ is a ball of radius $\frac{1}{4}$! Actually, the whole metric space $\mathbb{Z}_p$ is a ball of radius 1 (and is a p-adic analog of a real unit interval). For those who is familiar with functional analysis we mention also that the space $\mathbb{Z}_p$ is *complete* with respect to the p-adic distance (metric) d_p, and *compact*.

4. Elements of applied 2-adic analysis

The main goal of this section is to provide some experience in Calculus on $\mathbb{Z}_2$. We are not going to do this too formally since there are a number of excellent books and monographs on p-adic analysis, e.g. [24, 20, 17, 12]. We rather focus on those functions and techniques that later in the course will be used in our cryptographic applications, stream cipher design.

4.1. T-functions revisited.

We start with 2-adic extensions of what we called 'basic instructions'. These are primarily arithmetic operations (addition, subtraction, multiplication) and bitwise logical operations. These two set of operations are not mutually independent, some of them could be expressed via others. The following identities could be proved: For all

$u, v \in \mathbb{Z}_2$

$$\begin{aligned}
&\mathsf{NOT}(u) = u \,\mathsf{XOR}(-1);\\
&\mathsf{NOT}(u) + u = -1;\\
&u \,\mathsf{XOR}\, v = u + v - 2(u \,\mathsf{AND}\, v);\\
&u \,\mathsf{OR}\, v = u + v - (u \,\mathsf{AND}\, v);\\
&u \,\mathsf{OR}\, v = (u \,\mathsf{XOR}\, v) + (u \,\mathsf{AND}\, v).
\end{aligned} \tag{4.0.2}$$

During the course we often write $\oplus$ instead of XOR, also $\odot$, & or $\wedge$ instead of AND, and $\vee$ instead of OR. These operations (with the only exception of NOT) are functions of two 2-adic variables. To work with these functions we need to define 2-adic metric on a Cartesian square $\mathbb{Z}_2^2$. Having already defined metric on $\mathbb{Z}_2$ we define metric on a Cartesian product $\mathbb{Z}_2^n = \underbrace{\mathbb{Z}_2 \times \cdots \times \mathbb{Z}_2}_{n \text{ times}}$ in a standard manner: For $\mathbf{a} = (a^{(1)}, \dots, a^{(n)}), \mathbf{b} = (b^{(1)}, \dots, b^{(n)}) \in \mathbb{Z}_2^n$ we put $\|\mathbf{a}\|_2 = \max\{\|a^{(1)}\|_2, \dots, \|a^{(n)}\|_2\}$ and, respectively, $d_2(\mathbf{a}, \mathbf{b}) = \max\{d_2(a^{(1)}, b^{(1)}), \dots, d_2(a^{(n)}, b^{(n)})\}$. We also write $\mathbf{a} \equiv \mathbf{b} \pmod{2^i}$ whenever $a^{(j)} \equiv b^{(j)} \pmod{2^i}$ for all $j = 1, 2, \dots, n$.

Now it is a right time to consider T-functions as 2-adic mappings. Actually (see Definition 2.1) we define T-function as a special mapping that puts into a correspondence to every sequence of columnar m-dimensional Boolean vectors certain sequence of n-dimensional columnar Boolean vectors. Now we can read these sequence not column after column, but as a row after a row, starting with a top one. Each this row is an infinite sequence of zeros and ones; that is, a 2-adic integer. Thus,

> *we can consider a T-function F from Definition 2.1 as a mapping from $\mathbb{Z}_2^m$ into $\mathbb{Z}_2^n$ such that $F(\mathbf{a}) \equiv F(\mathbf{b}) \pmod{2^i}$ whenever $\mathbf{a} \equiv \mathbf{b} \pmod{2^i}$.*

From this observation immediately follows a very important theorem:

Theorem 4.1. *T-functions are mappings from $\mathbb{Z}_2^m$ into $\mathbb{Z}_2^n$ that satisfy Lipschitz condition with a constant 1:*

$$\|F(\mathbf{a}) - F(\mathbf{b})\|_2 \le \|\mathbf{a} - \mathbf{b}\|_2$$

and vise versa, all mappings that satisfy this condition are T-functions.

Corollary 4.2. *All T-functions are continuous 2-adic functions.*[8]

These easy claims are a hint that 2-adic analysis could be useful in study of T-functions; of course, only of properties that are of 'analytic nature', which could be properly stated in terms of analysis; that is, in terms of limits, convergence, derivatives, etc. We have not stated still *what* are these properties of T-functions that are crucial for cryptography. Yet, when we state these properties a bit later, we see that fortunately they are of this 'analytic nature'.

By the way, the above observation reflects a very specific *algebraic* nature of T-functions. In general algebra, a *congruence* of an algebraic system is an equivalence relation which is preserved by all operations of this system; that

[8]Any function that satisfy Lipschitz condition with respect to a certain metric is continuous with respect to this metric. Prove this!

is, if replacing operands by equivalent elements the result of the operation is equivalent to the original one. A function defined on (and valuated in) the algebraic system is called *compatible* whenever this function preserves all congruences of this algebraic system. The *only* congruences of the ring $\mathbb{Z}_p$ are congruences modulo p^k for $k = 1, 2, \ldots$. Thus, *T-functions are merely compatible functions on the ring $\mathbb{Z}_2$*, so we start using the term 'compatible' along with (or instead of) the term 'T-function'.

Actually, 'T-function' just means 'compatible on the ring $\mathbb{Z}_2$', and many further results holds for functions that are compatible on $\mathbb{Z}_p$, p prime. A p-adic compatible function is the function that satisfies p-adic Lipschitz condition with a constant 1, and vise versa.

4.2. **More compatible functions.** We already know that arithmetic operations (addition, subtraction, and multiplication), as well as bitwise logical operations (XOR, AND, etc.) are T-functions (that is, compatible 2-adic functions). Obviously, a *composition of compatible functions is a compatible function*. Whence, natural examples of compatible functions are *polynomials* with p-adic integer coefficients. That is, all polynomials with integer coefficients *are* T-functions!

With some extra efforts one could prove also that some other 'natural' functions are also T-functions:

$$\begin{aligned}
&\textit{exponentiation, } \uparrow\colon\ (u,v) \mapsto u \uparrow v = (1 + 2\cdot u)^v;\ \text{in particular,}\\
&\textit{raising to negative powers, } u \uparrow (-r) = (1 + 2\cdot u)^{-r}, r \in \mathbb{N};\ \text{and}\\
&\textit{division, } / : u/v = u \cdot (v \uparrow (-1)) = \frac{u}{1 + 2\cdot v}.
\end{aligned} \tag{4.2.1}$$

That is, these functions are well defined on $\mathbb{Z}_2$, and satisfy 2-adic Lipschitz condition with a constant 1. Use of compositions of these functions with the above mentioned bitwise logical instruction results in very wild-lloking functions, like this one:

$$(1 + x) \,\mathsf{XOR}\, 4 \cdot \left(1 - 2 \cdot \frac{x \,\mathsf{AND}\, x^2 + x^3 \,\mathsf{OR}\, x^4}{3 - 4 \cdot (5 + 6x^5)^{x^6 \,\mathsf{XOR}\, x^7}}\right)^{7 - \frac{8x^8}{9 + 10x^9}}$$

Despite this function could be *easily* evaluated on every digital computer (since this function is continuous in a computer's 2-adic world), we do not insist on using it (and similar) functions in applications: Compositions of the above mentioned functions may not be of big importance for cryptography since their program implementations are usually slow, yet they are of theoretical interest and often arise in studies. The p-adic analogs of the above functions could be naturally defined (write p instead of 2).

It also worth notice here that $(1 + p \cdot v)^{-1} = \sum_{i=0}^{\infty} (-1)^{i+1} p^i v^i$, and the series in the right-hand part of this equality are convergent for every $v \in \mathbb{Z}_p$.

We can describe univariate T-functions in some general way. It turns out that each function $f\colon \mathbb{N}_0 \to \mathbb{Z}_p$ (or, respectively, $f\colon \mathbb{N}_0 \to \mathbb{Z}$) admits one and only one representation in the form of so-called *Mahler interpolation series*

$$f(x) = \sum_{i=0}^{\infty} a_i \binom{x}{i}, \tag{4.2.2}$$

where $\binom{x}{i} = \frac{x(x-1)\cdots(x-i+1)}{i!}$ for $i = 1, 2, \ldots$, and $\binom{x}{0} = 1$; $a_i \in \mathbb{Z}_p$ (respectively, $a_i \in \mathbb{Z}$), $i = 0, 1, 2, \ldots$.

If f is uniformly continuous on $\mathbb{N}_0$ with respect to p-adic distance, it can be uniquely expanded to a uniformly continuous function on $\mathbb{Z}_p$. Hence the interpolation series for f converges uniformly on $\mathbb{Z}_p$. The following is true: The series $f(x) = \sum_{i=0}^{\infty} a_i \binom{x}{i}$, $(a_i \in \mathbb{Z}_p,\ i = 0, 1, 2, \ldots\ )$ converges uniformly on $\mathbb{Z}_p$ iff $\overset{p}{\lim\limits_{i\to\infty}}\, a_i = 0$, where $\overset{p}{\lim}$ is a limit with respect to the p-adic distance; hence uniformly convergent series defines a uniformly continuous function on $\mathbb{Z}_p$.

The following theorem holds:

Theorem 4.3. *The function $f\colon \mathbb{Z}_p \to \mathbb{Z}_p$ represented by (4.2.2) is compatible if and only if*

$$a_i \equiv 0 \pmod{p^{\lfloor \log_p i \rfloor}}$$

for all $i = p, p+1, p+2, \ldots$. (Here and after for a real α we denote $\lfloor \alpha \rfloor$ an integral part of α, i.e., the nearest to α rational integer not exceeding α.)

4.3. **Derivatives modulo p^k.** In this subsection we generalize the main notion of Calculus, a derivative. By the definition, for $\mathbf{a} = (a_1, \ldots, a_n)$ and $\mathbf{b} = (b_1, \ldots, b_n)$ of $\mathbb{Z}_p^{(n)}$ the congruence $\mathbf{a} \equiv \mathbf{b} \pmod{p^s}$ means that $\|a_i - b_i\|_p \le p^{-s}$ (or, the same, that $a_i = b_i + c_i p^s$ for suitable $c_i \in \mathbb{Z}_p$, $i = 1, 2, \ldots, s$); that is $\|\mathbf{a} - \mathbf{b}\|_p \le p^{-s}$.

Definition 4.4 (Derivations modulo p^k)**.** A function

$$F = (f_1, \ldots, f_m)\colon \mathbb{Z}_p^n \to \mathbb{Z}_p^m$$

is called *differentiable modulo p^k* at the point $\mathbf{u} = (u_1, \ldots, u_n) \in \mathbb{Z}_p^n$ iff there exist a positive integer rational N and an $n \times m$ matrix $F'_k(\mathbf{u})$ over $\mathbb{Z}_p$ (which is called *the Jacobi matrix modulo p^k* of the function F at the point $\mathbf{u}$) such that for each positive rational integer $K \ge N$ and each $\mathbf{h} = (h_1, \ldots, h_n) \in \mathbb{Z}_p^n$ the inequality $\|\mathbf{h}\|_p \le p^{-K}$ implies a congruence

$$F(\mathbf{u} + \mathbf{h}) \equiv F(\mathbf{u}) + \mathbf{h} \cdot F'_k(\mathbf{u}) \pmod{p^{k+K}}. \tag{4.4.1}$$

In case $m = 1$ the Jacobi matrix modulo p^k is called a *differential modulo p^k*. In case $m = n$ a determinant of the Jacobi matrix modulo p^k is called the *Jacobian modulo p^k*. The entries of the Jacobi matrix modulo p^k are called *partial derivatives modulo p^k* of the function F at the point $\mathbf{u}$. A partial derivative (respectively, a differential) modulo p^k we sometimes denote as $\frac{\partial_k f_i(\mathbf{u})}{\partial_k x_j}$ (respectively, as $d_k F(\mathbf{u}) = \sum_{i=1}^{n} \frac{\partial_k F(\mathbf{u})}{\partial_k x_i} d_k x_i$).

It could be proved that whenever F is compatible, then, if F is differentiable modulo p^k at some point, the entries of the Jacobi matrix are *necessarily* p-adic integers (such functions are said to have *integer-valued* derivative).

Since the notion of function that is differentiable modulo p^k is of high importance in theory that follows, we discuss this notion in details. First of all, we compare this notion to a classical notion of differentiable function.

Compare to differentiability, the differentiability modulo p^k is a weaker restriction. As a matter of fact, in a univariate case ($m = n = 1$), definition 4.4 just yields that

$$\frac{F(\mathbf{u}+\mathbf{h}) - F(\mathbf{u})}{\mathbf{h}} \approx F'_k(\mathbf{u})$$

Note that this $\approx$ ('approximately') implies the following:

$\approx$ with *arbitrarily high* precision $\Rightarrow$ differentiability;

$\approx$ with precision *not worse than* p^{-k} $\Rightarrow$ differentiability mod p^k.

It is obvious that whenever a function is differentiable (and its derivative is a p-adic integer), it is differentiable modulo p^k for all $k = 1, 2, \ldots$, and in this case the derivative modulo p^k is just a *reduction* of a derivative modulo p^k (note that according to definition 4.4 partial derivatives modulo p^k are determined up to a summand that is 0 modulo p^k).

For functions with integer-valued derivatives modulo p^k the 'rules of derivation modulo p^k' have the same (up to congruence modulo p^k instead of equality) form as for classical derivations. For instance, if both functions $G\colon \mathbb{Z}_p^s \to \mathbb{Z}_p^n$ and $F\colon \mathbb{Z}_p^n \to \mathbb{Z}_p^m$ are differentiable modulo p^k at the points, respectively, $\mathbf{v} = (v_1, \ldots, v_s)$ and $\mathbf{u} = G(\mathbf{v})$, and their partial derivatives modulo p^k at these points are p-adic integers, then a composition $F \circ G\colon \mathbb{Z}_p^s \to \mathbb{Z}_p^m$ of these functions is uniformly differentiable modulo p^k at the point $\mathbf{v}$, all its partial derivatives modulo p^k at this point are p-adic integers, and $(F \circ G)'_k(\mathbf{v}) \equiv G'_k(\mathbf{v})F'_k(\mathbf{u}) \pmod{p^k}$.

By the analogy with classical case we can give the following

Definition 4.5. A function $F\colon \mathbb{Z}_p^n \to \mathbb{Z}_p^m$ is said to be *uniformly differentiable modulo p^k on $\mathbb{Z}_p^{(n)}$* iff there exists $K \in \mathbb{N}$ such that (4.4.1) holds simultaneously for all $\mathbf{u} \in \mathbb{Z}_p^n$ as soon as $\|h_i\|_p \le p^{-K}$, $(i = 1, 2, \ldots, n)$. The least such $K \in \mathbb{N}$ is denoted via $N_k(F)$.

It could be shown that all partial derivatives modulo p^k of a uniformly differentiable modulo p^k function F are periodic functions with period $p^{N_k(F)}$ (see [3, Proposition 2.12]). This in particular implies that each partial derivative modulo p^k could be considered as a function defined on the residue ring $\mathbb{Z}/p^{N_k(F)}\mathbb{Z}$ modulo $p^{N_k(F)}$. Moreover, if a continuation $\tilde{F}$ of the function $F = (f_1, \ldots, f_m)\colon \mathbb{N}_0^n \to \mathbb{N}_0^m$ to the space $\mathbb{Z}_p^n$ is uniformly differentiable modulo p^k on the $\mathbb{Z}_p^n$, then one could continue both the function F and all its (partial) derivatives modulo p^k to the space $\mathbb{Z}_p^n$ simultaneously. This implies that we could study if necessary (partial) derivatives modulo p^k of the function $\tilde{F}$ instead of studying those of F and vise versa. For example, a partial derivative $\frac{\partial_k f_i(\mathbf{u})}{\partial_k x_j}$ modulo p^k vanishes modulo p^k at no point of $\mathbb{Z}_p^n$ (that is, $\frac{\partial_k f_i(\mathbf{u})}{\partial_k x_j} \not\equiv 0 \pmod{p^k}$ for all $u \in \mathbb{Z}_p^n$, or, the same $\left\|\frac{\partial_k f_i(\mathbf{u})}{\partial_k x_j}\right\|_p > p^{-k}$ everywhere on $\mathbb{Z}_p^n$) if and only if $\frac{\partial_k f_i(\mathbf{u})}{\partial_k x_j} \not\equiv 0 \pmod{p^k}$ for all $u \in \{0, 1, \ldots, p^{N_k(F)} - 1\}$.

To calculate a derivative of, for instance, a T-function that is a composition of basic instructions one needs to know derivatives of these basic

instructions (i.e., arithmetic, bitwise logical, etc.) Thus, we briefly introduce a p-adic analog of a 'table of derivatives' of classical Calculus.

Examples 4.6. Derivatives of bitwise logical operations.

(1) *the function* $f(x) = x \,\mathsf{AND}\, c$ *is uniformly differentiable on* $\mathbb{Z}_2$ *for any* $c \in \mathbb{Z}$; $f'(x) = 0$ *for* $c \geq 0$, *and* $f'(x) = 1$ *for* $c < 0$, since $f(x+2^n s) = f(x)$, and $f(x+2^n s) = f(x)+2^n s$ for $n \geq l(|c|)$, where $l(|c|)$ is the bit length of absolute value of c (mind that for $c \geq 0$ the 2-adic representation of $-c$ starts with $2^{l(c)} - c$ in less significant bits followed by $\ldots 11$: $-1 = \ldots 11$, $-3 = \ldots 11101$, etc.).

(2) *the function* $f(x) = x \,\mathsf{XOR}\, c$ *is uniformly differentiable on* $\mathbb{Z}_2$ *for any* $c \in \mathbb{Z}$; $f'(x) = 1$ *for* $c \geq 0$, *and* $f'(x) = -1$ *for* $c < 0$. This immediately follows from (1) since $u \,\mathsf{XOR}\, v = u + v - 2(x \,\mathsf{AND}\, v)$ (see (4.0.2)); thus $(x \,\mathsf{XOR}\, c)' = x' + c' - 2(x \,\mathsf{AND}\, c)' = 1 + 2 \cdot (0,$ for $c \geq 0$; or $-1,$ for $c < 0)$.

(3) in the same manner it could be shown that *functions* $(x \bmod 2^n) = x \,\mathsf{AND}(2^n - 1)$ (a reduction modulo 2^n), $\mathsf{NOT}(x)$ *and* $(x \,\mathsf{OR}\, c)$ *for* $c \in \mathbb{Z}$ *are uniformly differentiable on* $\mathbb{Z}_2$, *and* $(x \bmod 2^n)' = 0$, $(\mathsf{NOT}\, x)' = -1$, $(x \,\mathsf{OR}\, c)' = 1$ *for* $c \geq 0$, $(x \,\mathsf{OR}\, c)' = 0$ *for* $c < 0$.

(4) *the function* $f(x,y) = x \,\mathsf{XOR}\, y$ *is not uniformly differentiable on* $\mathbb{Z}_2^2$, *yet it is uniformly differentiable modulo* 2 *on* $\mathbb{Z}_2^2$; from (2) it follows that its partial derivatives modulo 2 are 1 everywhere on $\mathbb{Z}_2^2$.

Here how it works altogether:

Example. The function $f(x) = x + (x^2 \,\mathsf{OR}\, 5)$ is uniformly differentiable on $\mathbb{Z}_2$, and $f'(x) = 1 + 2x \cdot (x \,\mathsf{OR}\, 5)' = 1 + 2x$.

The function $F(x,y) = (f(x,y), g(x,y)) = (x \oplus 2(x \wedge y), (y + 3x^3) \oplus x)$ is uniformly differentiable modulo 2 as bivariate function, and $N_1(F) = 1$; namely

$$F(x+2^n t, y+2^m s) \equiv F(x,y) + (2^n t, 2^m s) \cdot \begin{pmatrix} 1 & x+1 \\ 0 & 1 \end{pmatrix} \pmod{2^{k+1}}$$

for all $m, n \geq 1$ (here $k = \min\{m,n\}$). The matrix $\begin{pmatrix} 1 & x+1 \\ 0 & 1 \end{pmatrix} = F'_1(x,y)$ is Jacobi matrix modulo 2 of F; here how we calculate partial derivatives modulo 2: for instance, $\frac{\partial_1 g(x,y)}{\partial_1 x} = \frac{\partial_1 (y+3x^3)}{\partial_1 x} \cdot \frac{\partial_1 (u \oplus x)}{\partial_1 u}\big|_{u=y+3x^3} + \frac{\partial_1 x}{\partial_1 x} \cdot \frac{\partial_1 (u \oplus x)}{\partial_1 x}\big|_{u=y+3x^3} = 9x^2 \cdot 1 + 1 \cdot 1 \equiv x + 1 \pmod 2$. Note that a partial derivative modulo 2 of the function $2(x \wedge y)$ is always 0 modulo 2 because of the multiplier 2: The function $x \wedge y$ is not differentiable modulo 2 as bivariate function, yet $2(x \wedge y)$ is. So the Jacobian of the function F is $\det F'_1 \equiv 1 \pmod 2$.

Now let $F = (f_1, \ldots, f_m) \colon \mathbb{Z}_p^n \to \mathbb{Z}_p^m$ and $f \colon \mathbb{Z}_p^n \to \mathbb{Z}_p$ be compatible functions, which are uniformly differentiable on $\mathbb{Z}_p^n$ modulo p. This is a relatively weak restriction since all uniformly differentiable on $\mathbb{Z}_p^n$ functions, as well as functions, which are uniformly differentiable on $\mathbb{Z}_p^n$ modulo p^k for some $k \geq 1$, are uniformly differentiable on $\mathbb{Z}_p^n$ modulo p; note that $\frac{\partial F}{\partial x_i} \equiv \frac{\partial_k F}{\partial_k x_i} \equiv \frac{\partial_{k-1} F}{\partial_{k-1} x_i} \pmod{p^{k-1}}$. Moreover, as it was mentioned, all values

of all partial derivatives modulo p^k (and thus, modulo p) of F and f are p-adic integers everywhere on, respectively, $\mathbb{Z}_p^n$ and $\mathbb{Z}_p$, so to calculate these values one can use the techniques considered above.

5. Stream ciphers and 2-adic ergodic theory

In this section we discuss what conditions state update and output functions of a pseudorandom generator should satisfy to guarantee some crucial cryptographic properties of the produced sequence. It turns out that whenever these functions are T-functions, the properties are tightly connected with the behaviour of the functions with respect to a natural probabilistic measure on the space $\mathbb{Z}_2$. We start with defining this measure.

5.1. **Notions of p-adic dynamics.** When we measure a square of a figure on a plane (or a volume of a body in a space), we associate a real number to the figure (resp., to the body). These are natural examples of *measures*. We are not going to recall basic notions of measure theory here, referring to any book on this topic. We only mention that we could define a measure μ on some set $\mathbb{S}$ by assigning non-negative real numbers to some subsets that are called elementary. All other *measurable* subsets are compositions of these elementary subsets with respect to countable unions, intersections, and complements. Actually, if a measurable subset $S \subset \mathbb{S}$ is a disjoint union of elementary measurable subsets E_j, $S = \cup_{j=0}^{\infty} E_j$, then $\mu(S) = \sum_{j=0}^{\infty} \mu(E_j)$, and the series in the right-hand part must be convergent. The set $\mathbb{S}$ with so defined measure μ is called a *measurable space*.

The elementary subsets in $\mathbb{Z}_p$ are balls $B_{p^{-k}}(a) = a + p^k\mathbb{Z}_p$. To each such ball we assign a number $\mu_p(B_{p^{-k}}(a)) = \frac{1}{p^k}$. It could be verified that we indeed define a measure on the space $\mathbb{Z}_p$, and this measure is a probabilistic measure, $\mu_p(\mathbb{Z}_p) = 1$. This measure μ_p is called a (normalized) *Haar measure* on $\mathbb{Z}_p$.

We say that we have a *dynamical system* on a measurable space $\mathbb{S}$, whenever we consider a triple $(\mathbb{S}; \mu; f)$, where $\mathbb{S}$ is a measurable space with measure μ, and $f\colon \mathbb{S} \to \mathbb{S}$ is a *measurable function*; that is, an f-preimage of every measurable subset is a measurable subset. Dynamical system theory is a reach mathematical theory which is applied in different parts of science and industry. As a matter of fact, in this course we will discuss *applications of 2-adic dynamical systems theory to stream cipher design*.

A *trajectory* of a dynamical system is a sequence

$$x_0, x_1 = f(x_0), \dots, x_i = f(x_{i-1}) = f^i(x_0), \dots$$

of points of the space $\mathbb{S}$, x_0 is called an *initial* point of the trajectory. If $F\colon \mathbb{S} \to \mathbb{T}$ is a measurable mapping to some other measurable space $\mathbb{T}$ with a measure ν (that is, an F-preimage of any ν-measurable subset of $\mathbb{T}$ is a μ-measurable subset of X), the sequence $F(x_0), F(x_1), F(x_2), \dots$ is called an *observable*. Note that the trajectory formally looks like the sequence of states of a pseudorandom generator, whereas the observable resembles the output sequence, cf. subsection 2.1. Further we will see that is not just an analogy.

The two important notions of dynamical systems theory follow: A mapping $F\colon \mathbb{S} \to \mathbb{Y}$ of a measurable space $\mathbb{S}$ into a measurable space $\mathbb{Y}$ endowed

with probabilistic measure μ and ν, respectively, is said to be *measure-preserving* (or, sometimes, *equiprobable*) whenever $\mu(F^{-1}(S)) = \nu(S)$ for each measurable subset $S \subset \mathbb{Y}$. In case $\mathbb{S} = \mathbb{Y}$ and $\mu = \nu$, a measure-preserving mapping F is said to be *ergodic* whenever for each measurable subset S such that $F^{-1}(S) = S$ holds either $\mu(S) = 1$ or $\mu(S) = 0$. Loosely speaking, any invariant set of the ergodic mapping is either nothing, or everything.

The p-adic ergodic theory studies ergodic (with respect to the Haar measure) transformations of the space of p-adic numbers, conditions that provide ergodicity, etc. It is a rapidly developing mathematical theory, with various applications, see e.g. [13]. Actually, as we will see, the course is a development of p-adic ergodic theory with special interest to pseudorandom number generators (particulary, stream ciphers).[9] And now it is a right time to discuss how the above notions are related to properties of pseudorandom generators.

5.2. **What is a good PRNG.** A PRNG which could be considered any good obviously must meet the following conditions:

- The output sequence must be pseudorandom (i.e., must pass certain statistical tests).
- For cryptographic applications, given a segment $z_j, z_{j+1}, \dots, z_{j+s-1}$ of the output sequence, finding the corresponding initial state (which usually is a key) must be infeasible in some properly defined sense.
- The PRNG must be suitable for software (or hardware) implementation; the performance must be sufficiently fast.

In case the PRNG is an automaton described by Figure 1 we could re-state these conditions as follows:

First of all, we state

> Condition 1: The state update function f must provide pseudorandomness; in particular, it must guarantee *uniform distribution* and *long period* of the state update sequence $\{u_i\}$.

It would be great if this sequence is secure; that is, given u_i, it is infeasible neither to find (or to predict) u_{i+1}, nor to find u_0. Unfortunately, this is not easy to provide these properties: Generators that are 'provably secure', that is, supplied with proofs (which are based on some plausible, yet still unproven conjectures) that their output sequences can not be predicted by polynomial-time algorithms, are too slow for most practical applications. In real life one has to undertake additional efforts to make the algorithm secure. Usually this could be achieved with the use of the output function. Thus, we need

> Condition 2: The output function F must not spoil pseudorandomness (at least, the output sequence $\{z_i\}$ must be *uniformly distributed* and must have *long period*).
>
> Moreover, in cryptographic applications the function F must make the PRNG secure: (in particular, given z_i, it must be *difficult to find u_i from the equation $z_i = F(u_i)$*).

[9] By the way, methods developed within this approach could be applied to solve some problems of p-adic ergodic theory, see [1]

Finally, we can formulate

> Condition 3: To make the PRNG any suitable for software/hardware implementations, *both f and G must be compositions of basic processor instructions.*

In section 2 we already have discussed how one could satisfy condition 3: It is sufficient to choose both f and F from the class of T-functions. Thus, we can assume that $f\colon \mathbb{Z}/2^n\mathbb{Z} \to \mathbb{Z}/2^n\mathbb{Z}$ and $F\colon \mathbb{Z}/2^n\mathbb{Z} \to \mathbb{Z}/2^m\mathbb{Z}$ (usually, $m \le n$).

Now, to satisfy condition 1, one could take the state update function $f\colon \mathbb{Z}/2^n\mathbb{Z} \to \mathbb{Z}/2^n\mathbb{Z}$ with a *single cycle property*; that is, f permutes elements of $\mathbb{Z}/2^n\mathbb{Z}$ cyclically.

The state update sequence

$$u_0,\ u_1 = f(u_0), \dots, u_{i+1} = f(u_i) = f^{i+1}(u_0), \dots$$

of n-bit words will have then the *longest possible period* (of length 2^n), and *strict uniform distribution*; that is, each n-bit word will occur at the period exactly once.

To satisfy the first part of condition 2, one could take the output function $F\colon \mathbb{Z}/2^n\mathbb{Z} \to \mathbb{Z}/2^m\mathbb{Z}$ to be *balanced*: That is, to each m-bit word the mapping F maps the same number of n-bit words (that's why $m \le n$). For $m = n$ balanced mappings are just *invertible* (that is, bijective, one-to-one) mappings. Obviously, if a balanced output function is applied to a strictly uniformly distributed sequence of states, the output sequence (of m-bit words) is also strictly uniformly distributed: It is *periodic with a period of length 2^n, and each m-bit word occurs at the period exactly 2^{n-m} times.*

For $m \ll n$, balanced functions could serve us to satisfy the second part of condition 2, since the equation $y_i = G(x_i)$ has too many solutions then, 2^{n-m} (so it is infeasible to an attacker to try them all).

Thus, we must know how to construct balanced (or single-cycle) functions out of basic processor instructions. This is where the non-Archimedean analysis comes into play!

5.3. **A bridge.** Now we make our studies more formal. Let $F\colon \mathbb{Z}_p^n \to \mathbb{Z}_p^m$ be a compatible function; that is, let F satisfy the p-adic Lipschitz condition with a constant 1 (see section 4). In other words, for every $k = 1, 2, \dots$, and for every $\mathbf{a}, \mathbf{b} \in \mathbb{Z}_p^n$, $F(\mathbf{a}) \equiv F(\mathbf{b}) \pmod{p^k}$ whenever $\mathbf{a} \equiv \mathbf{b} \pmod{p^k}$ (see subsection 4.3 for the definition of $\bmod p^k$). This means that, given a compatible mapping $F\colon \mathbb{Z}_p^n \to \mathbb{Z}_p^m$, its *reduction* $F \bmod p^k$ *modulo* p^k is a *well defined mapping*

$$F \bmod p^k\colon (\mathbb{Z}/p^k\mathbb{Z})^n \to (\mathbb{Z}/p^k\mathbb{Z})^m$$

of respective Cartesian powers of the residue ring $\mathbb{Z}/p^k\mathbb{Z}$. We call the mapping $F \bmod p^k$ the *induced* mapping. The idea is quite clear: Reduction modulo p^k just deletes all most significant digits (starting with the k-th digit) both of arguments and of values of the function F.

Definition 5.1. A compatible mapping $F\colon \mathbb{Z}_p \to \mathbb{Z}_p$ is said to be *bijective* (resp., *transitive*) modulo p^k iff the induced mapping $x \mapsto F(x) \pmod{p^k}$ is a (single-cycle) permutation of the elements of the ring $\mathbb{Z}/p^k\mathbb{Z}$.

Balance modulo p^k could be defined by an analogy. Now we can state the central result of this section:

Theorem 5.2 (see [5]). *For $m = n = 1$, a compatible mapping $F\colon \mathbb{Z}_p^n \to \mathbb{Z}_p^m$ preserves the normalized Haar measure μ_p on $\mathbb{Z}_p$ (resp., is ergodic with respect to μ_p) if and only if it is bijective (resp., transitive) modulo p^k for all $k = 1, 2, 3, \ldots$.*

For $n \geq m$, the mapping F preserves the measure μ_p if and only if it induces a balanced mapping of $(\mathbb{Z}/p^k\mathbb{Z})^n$ onto $(\mathbb{Z}/p^k\mathbb{Z})^m$, for all $k = 1, 2, 3, \ldots$.

This theorem acts like a bridge between p-adic ergodic theory and stream cipher design: We consider the corresponding PRNG as approximation with respect to 2-adic metric of some ergodic dynamical system on 2-adic integers. In a pseudorandom generator, we can take compatible ergodic functions for state update functions; also we can take compatible measure-preserving functions for output functions. The reduction modulo 2^n a computer performs automatically. In particular, for $p = 2$ from theorem 5.2 we obtain:

- measure preservation = invertibility modulo 2^k for all $k \in \mathbb{N}$;
- in dimensions > 1, i.e., for $F\colon \mathbb{Z}_2^n \to \mathbb{Z}_2^m$,
  measure preservation = balance modulo 2^k for all $k \in \mathbb{N}$;
- ergodicity = single cycle property modulo 2^k for all $k \in \mathbb{N}$.

In other words, a compatible function $F\colon \mathbb{Z}_2 \to \mathbb{Z}_2$ is measure-preserving (respectively, ergodic) if and only if the corresponding T-function $F \pmod{2^n}$ on n-bit words (which is merely an approximation of F with precision $\frac{1}{2^n}$) is invertible or, respectively, has a single cycle property!

Now the problem is how to describe these measure-preserving (in particular, ergodic) mappings in the class of all compatible mappings. We start to develop some theory to answer the following questions: What compositions of basic instructions are measure-preserving? are ergodic? Given a composition of basic instructions, is it measure-preserving? is it ergodic?

6. Tools

The main goal of this section is to describe some tools with the use of which we could answer the above stated questions. However, we start with some historical observations.

6.1. **A phenomenon.** Study of pseudorandom generators has a long history. You can read about this issue in, for instance, an excellent book of Donald Knuth [16]. Here we discuss briefly a short passage of this long story, aiming to make some important observations.

One could notice that behavior of a mapping modulo p^N, where N is big, is totally determined by the behavior of this mapping modulo p^n, where n is small. One of the first generators that demonstrate this behaviour is

Linear Congruential Generator (Hull and Dobell, 1962):

The mapping

$$x \mapsto a \cdot x + b \pmod{p^N},$$

where $a, b \in \mathbb{Z}$, $N \geq 2$, is a permutation with a single cycle property if and only if $x \mapsto a \cdot x + b \pmod{p^n}$ is a permutation with a single cycle property for $n = 1$ in case p odd, or for $n = 2$, otherwise.

The following important example is

Bijectivity Criterion for Polynomials with Integer Coefficients (proved and re-proved by a number of authors; known since 1960th):

The mapping

$$x \mapsto f(x) \pmod{p^N},$$

where $N \geq 2$ and f is a polynomial with rational integer coefficients, is bijective if and only if $x \mapsto f(x) \pmod{p^n}$ is bijective for $n = 2$.

Yet another one example:

Quadratic Generator (Coveyou, 1969):

The mapping

$$x \mapsto f(x) \pmod{p^N},$$

where $N \geq 3$ and f is a quadratic polynomial with rational integer coefficients, is a permutation with a single cycle property iff $x \mapsto f(x) \pmod{p^n}$ is a permutation with a single cycle property for $n = 3$ in case $p \in \{2, 3\}$, or for $n = 2$, otherwise.

It worth notice here that in 1980th M. V. Larin proved that the *word 'quadratic' in the statement could be omitted*! The result was spread as a manuscript that time, a journal publication [18] appeared much later.

6.2. **Explanation: p-adic derivations.** Looking at the examples of the preceding subsection, we naturally start suspecting that some very strong reason for such behaviour must exist! The following theorem, which was published in 1993 [4, 3], gives an explanation:

Theorem 6.1. *Let a compatible function $F\colon \mathbb{Z}_p \to \mathbb{Z}_p$ be uniformly differentiable modulo p^2. Then F is ergodic if and only if it is transitive modulo $p^{N_2(F)+1}$ for odd prime p or, respectively, modulo $2^{N_2(F)+2}$ for $p = 2$.*

This theorem works for a much wider class of functions that the ones mentioned in the above examples. Actually, this class includes functions that are compositions of not exceptionally arithmetic operations, but of logical operations as well. To illustrate the techniques, consider the following example.

Example 6.2. In their paper [15] of 2002 Klimov and Shamir write that

> ...neither the invertibility nor the cycle structure of $x+(x^2 \vee 5)$ could be determined by his (*i.e., mine — V.A.*) techniques.

See however how it could be immediately done with the use of Theorem 6.1: The function $f(x) = x + (x^2 \vee 5)$ is uniformly differentiable on $\mathbb{Z}_2$, thus, it is uniformly differentiable modulo 4 (see 4.6 and an example thereafter), and $N_2(f) = 3$. Indeed, $(x + h) \,\mathsf{OR}\, 5 = (x \,\mathsf{OR}\, 5) + h$ whenever $h \equiv 0 \pmod 8$ (the latter congruence is obvious since the base-2 expansion of 5 is ...000101).

Now to prove that f is ergodic, in view of 6.1 it suffices to demonstrate that f induces a permutation with a single cycle on $\mathbb{Z}/32$. Direct calculations show that the string

$$0, f(0) \bmod 32, f^2(0) \bmod 32 = f(f(0)) \bmod 32, \ldots, f^{31}(0) \bmod 32$$

is a permutation of the string $0, 1, 2, \ldots, 31$, thus ending the proof.

In connection with Theorem 6.1, the following natural question arises: What about ergodicity in higher dimensions? Unfortunately, for uniformly differentiable modulo p function the answer is negative. The following result could be considered as a non-existence theorem for compatible smooth ergodic mappings in higher dimensions.

Theorem 6.3 (see [4, 3])**.** *Let the function $F = (f_1, \ldots, f_n)\colon \mathbb{Z}_p^n \to \mathbb{Z}_p^n$ be compatible, ergodic, and uniformly differentiable modulo p on $\mathbb{Z}_p$. Then $n = 1$.*

Note. Non-differentiable mod$p$ ones do exist for $n > 1$

The following theorem, which uses derivations modulo p instead of p^2, could be applied to construct balanced mappings to serve as output functions of PRNG.

Theorem 6.4 (see [5])**.** *Let $F\colon \mathbb{Z}_p^n \to \mathbb{Z}_p^m$ be a compatible function that is uniformly differentiable modulo p. Then F preserves measure whenever it is balanced modulo p^k for some $k \ge N_1(F)$ and the rank of its Jacobi matrix $F_1'(u)$ modulo p is exactly m at all points $\mathbf{u} = (u_1, \ldots, u_n) \in (\mathbb{Z}/p^k)^n$.*

Proof. For $\xi \in (\mathbb{Z}/p^s)^m$ denote

$$F_s^{-1}(\xi) = \{\gamma \in (\mathbb{Z}/p^s)^n \colon F(\gamma) \equiv \xi \pmod{p^s}\}.$$

Let $s \ge k \ge N_1(F)$. Since F is compatible, and hence F is a sum of a compatible function and a periodic function with period $p^{N_1(F)}$ (see 2.10 of [3]), we conclude that if $\eta \in F_{s+1}^{-1}(\xi)$, then $\bar\eta \in F_s^{-1}(\bar\xi)$. Here and further we denote via $\bar\alpha = (\bar\alpha_1, \ldots, \bar\alpha_m) \in (\mathbb{Z}/p^s)^m$ the residue modulo p^s, $\alpha \bmod p^s = (\alpha_1 \bmod p^s, \ldots, \alpha_m \bmod p^s)$, where $\alpha = (\alpha_1, \ldots, \alpha_m) \in (\mathbb{Z}/p^{s+1})^m$.

Put $\lambda = \bar\eta + p^s\sigma \in (\mathbb{Z}/p^{s+1})^n$, where $\sigma \in (\mathbb{Z}/p)^n$. In view of the uniform differentiability of the function F modulo p (see 4.4), we have

$$F(\lambda) \equiv F(\eta) + p^s \sigma F_1'(\bar\eta) \pmod{p^{s+1}}. \tag{6.4.1}$$

Since $F(\bar\eta) \equiv \bar\xi + p^k\beta \pmod{p^{s+1}}$ and $\xi = \bar\xi + p^s\gamma$ for suitable $\beta, \gamma \in (\mathbb{Z}/p)^{(m)}$, in view of (6.4.1) we conclude that $\lambda \in F_{s+1}^{-1}(\xi)$ if and only if $\bar\lambda \in F_s^{-1}(\xi)$ (i.e., $\bar\eta \in F_s^{-1}(\xi)$) and α satisfies the following system of linear equations over a finite field $\mathbb{Z}/p$:

$$\beta + \alpha F_1'(\bar\eta) = \gamma. \tag{6.4.2}$$

Thus, if columns of the matrix $F_1'(\bar\eta)$ are linearly independent over $\mathbb{Z}/p$, then linear system (6.4.2) has exactly p^{n-m} pairwise distinct solutions for arbitrary $\beta, \gamma \in (\mathbb{Z}/p)^{(m)}$. From here it follows that

$$|F_{s+1}^{-1}(\xi)| = |F_s^{-1}(\xi)| p^{n-m}. \tag{6.4.3}$$

Hence, if F is equiprobable modulo p^k (i.e., if $|F_s^{-1}(\bar\xi)|$ does not depend on $\bar\xi$) and if rank of the matrix $F_1'(\bar\eta)$ is m, then (6.4.3) implies that F is balanced modulo p^{s+1}. □

Corollary 6.5. *Under assumptions of theorem 6.4:*

- *If $m = 1$, then F is measure-preserving whenever F is balanced modulo p^k for some $k \ge N_1(F)$, and the differential d_1F modulo p of the function F vanishes at no point of $(\mathbb{Z}/p^k|Z)^n$.*

- *Let $f(x_1,\ldots,x_n)$ be a polynomial in variables $x_1,\ldots,x_n$, and let all coefficients of f are p-adic integers. The polynomial f preserves measure whenever it is balanced modulo p and all its partial derivatives vanishes simultaneously modulo p at no point of $(\mathbb{Z}/p\mathbb{Z})^n$* (i.e., are simultaneously congruent to 0 modulo p nowhere) on $(\mathbb{Z}/p\mathbb{Z})^n$.

For $m=n$ the above stated sufficient conditions of measure preservation becomes also necessary ones.

Theorem 6.6. *A compatible and uniformly differentiable modulo p function*

$$F=(f_1,\ldots,f_m)\colon \mathbb{Z}_p^n\to\mathbb{Z}_p^n$$

preserves measure if and only if it is bijective modulo $p^{N_1(F)}$ and its Jacobian modulo p vanishes at no point of $(\mathbb{Z}/p^{N_1(F)}\mathbb{Z})^n$ (Equivalent condition: If and only if F is bijective modulo $p^{N_1(F)+1}$).

Proof. If F is bijective modulo $p^{N_1(F)}$, and if its Jacobian modulo p vanishes nowhere on $(\mathbb{Z}/p^{N_1(F)})^n$, then in view of Theorem 6.4 F preserves measure.

Vise versa, let F preserve measure, i.e., let F be bijective modulo p^k for all $k\ge N$, where N is some positive rational integer. Now take $k\ge\max\{N,N_1(F)\}$, then the definition of uniform differentiability modulo p implies that

$$F(u+p^k\alpha)\equiv F(u)+p^k\alpha F_1'(u)\pmod{p^{k+1}} \tag{6.6.1}$$

for all $u,\alpha\in\mathbb{Z}_p$. Here $F_1'(u)$ is an $n\times n$ matrix over a field $\mathbb{Z}/p$. If $\det F_1'(u)\equiv 0\pmod p$ for some $u\in\mathbb{Z}_p^n$ (or, the same, for some $u\in\{0,1,\ldots,p^{N_1(F)}-1\}^n$ in view of the periodicity of partial derivatives modulo p), then there exists $\alpha\in\{0,1,\ldots,p-1\}^n,\alpha\not\equiv(0,\ldots,0)\pmod p$, such that $\alpha F_1'(u)\equiv(0,\ldots,0)\pmod p$. But then (6.6.1) implies that $F(u+p^k\alpha)\equiv F(u)\pmod{p^{k+1}}$. The latter contradicts the bijectivity modulo p^{k+1} of the function F, since for $u\in\{0,1,\ldots,p^{N_1(F)}-1\}^n$ we have $u,u+p^k\alpha\in\{0,1,\ldots,p^{k+1}-1\}^n$ and $u+p^k\alpha\ne u$.

Now we prove the criterion in the equivalent form. Let F be bijective modulo $p^{N_1(F)}$. Then assuming $k=N_1(F)$ in the above argument, we conclude that $\det F_1'(u)\not\equiv 0\pmod p$ for all $u\in\mathbb{Z}_p^n$. According to Theorem 6.4, this implies that F preserves measure.

Let F preserve measure, and let F be not bijective modulo p^k for some $k\ge N_1(F)$. We prove that in this case F is not bijective modulo p^{k+1}.

Choose $u,v\in\{0,1,\ldots,p^k-1\}^n$ such that $u\ne v$ $F(u)\equiv F(v)\pmod{p^k}$. Then either $F(u)\equiv F(v)\pmod{p^{k+1}}$ (i.e., F is not bijective modulo p^{k+1}), or $F(u)\not\equiv F(v)\pmod{p^{k+1}}$. Yet in the latter case we have $F(u)\equiv F(v)+p^k\alpha\pmod{p^{k+1}}$ for some $\alpha\in\{0,1,\ldots,p-1\}^n$, $\alpha\not\equiv(0,\ldots,0)\pmod p$. Consider $u_1=u+p^k\beta$, where $\beta\in\{0,1,\ldots,p-1\}^n$ with $\beta\not\equiv(0,\ldots,0)\pmod p$ and $\beta F_1'(u)+\alpha\equiv(0,\ldots,0)\pmod p$. Such β exists, since F preserves measure and, consequently, $\det F_1'(u)\not\equiv 0\pmod p$, as this have been proven already. Now the definition of uniform differentiability modulo p implies that

$$F(u+p^k\beta)\equiv F(u)+p^k\beta F_1'(u)\equiv F(v)+p^k\alpha+p^k\beta F_1'(u)\equiv F(v)\pmod{p^{k+1}}, \tag{6.6.2}$$

where $u+p^k\beta \in \{0,1,\dots,p^{k+1}-1\}^{(n)}$ and $u+p^k\alpha \neq v$ (since $u \neq v$). Thus (6.6.2) in combination with our assumption imply that F is not bijective modulo p^{k+1}. Applying this argument sufficient number of times, we conclude that F is not bijective modulo p^s for all $s \geq k$. But at the same time F preserves measure. A contradiction. $\square$

Comparing theorems 6.4 and 6.6 one may ask whether sufficient conditions of theorem 6.4 are also necessary. The answer is negative: In [5] it is proved that the function $f(x,y)=2x+y^3$ on $\mathbb{Z}_2$ provides a counter-example.

Open question. Characterize all compatible measure-preserving mappings

$$F=(f_1,\dots,f_m)\colon \mathbb{Z}_p^n \to \mathbb{Z}_p^n$$

with $m<n$. The answer is not known even under restriction that all f_i are polynomials over $\mathbb{Z}_p$.

The technique presented in this subsection is rather effective: Actually, all the examples of preceding subsection could be deduced from the results of this subsection. Moreover, all results of [15] also could be proved by these techniques. We re-prove these results to illustrate our techniques:

Examples 6.7. The following is true:

(1) *A mapping*

$$(x,y)\mapsto F(x,y)=(x\oplus 2(x\wedge y),(y+3x^3)\oplus x) \bmod 2^r$$

of $(Z/2^r)^2$ *onto* $(Z/2^r)^2$ *is bijective for all* $r=1,2,\dots$

Indeed, the function F is bijective modulo $2^{N_1(F)}=2$ (direct verification) and $\det(F_1'(\mathbf{u}))\equiv 1 \pmod 2$ for all $\mathbf{u}\in(\mathbb{Z}/2)^2$ (see 4.6 and example thereafter).

(2) *The following mappings of* $\mathbb{Z}/2^r$ *onto* $\mathbb{Z}/2^r$ *are bijective for all* $r=1,2,\dots$:

$$\begin{array}{lcl} x\mapsto & (x+2x^2) & \mathrm{mod} 2^r,\\ x\mapsto & (x+(x^2\vee 1)) & \mathrm{mod} 2^r,\\ x\mapsto & (x\oplus(x^2\vee 1)) & \mathrm{mod} 2^r. \end{array}$$

Indeed, all three mappings are uniformly differentiable modulo 2, and $N_1=1$ for all of them. So it suffices to prove that all three mappings are bijective modulo 2, i.e. as mappings of the residue ring $\mathbb{Z}/2$ modulo 2 onto itself (this could be checked by direct calculations), and that their derivatives modulo 2 vanish at no point of $\mathbb{Z}/2$. The latter also holds, since the derivatives are, respectively,

$$\begin{array}{lcll} 1+4x & \equiv & 1 & \pmod 2,\\ 1+2x\cdot 1 & \equiv & 1 & \pmod 2,\\ 1+2x\cdot 1 & \equiv & 1 & \pmod 2, \end{array}$$

since $(x^2\vee 1)'=2x\cdot 1\equiv 1 \pmod 2$, and $(x\oplus C)_1'\equiv 1 \pmod 2$, (see 4.6).

(3) *The following closely related variants of the previous mappings of* $\mathbb{Z}/2^r$ *onto* $\mathbb{Z}/2^r$ *are not bijective for all* $r=1,2,\dots$:

$$\begin{array}{lcl} x\mapsto & (x+x^2) & \mathrm{mod} 2^r,\\ x\mapsto & (x+(x^2\wedge 1)) & \mathrm{mod} 2^r,\\ x\mapsto & (x+(x^3\vee 1)) & \mathrm{mod} 2^r, \end{array}$$

since they are compatible but not bijective modulo 2.

(4) (see [21], also [15, Theorem 1]) *Let* $P(x) = a_0 + a_1x + \cdots + a_dx^d$ *be a polynomial with integral coefficients. Then* $P(x)$ *is a permutation polynomial* (i.e., is bijective) *modulo* 2^n, $n > 1$ *if and only if* a_1 *is odd,* $(a_2 + a_4 + \cdots)$ *is even, and* $(a_3 + a_5 + \cdots)$ *is even.*

In view of 6.6 we must verify whether the two conditions hold: first, whether P is bijective modulo 2, and second, whether $P'(z) \equiv 1 \pmod 2$ for $z \in \{0, 1\}$. The first condition implies that $P(0) = a_0$ and $P(1) = a_0 + a_1 + a_2 + \cdots a_d$ must be distinct modulo 2; hence $a_1 + a_2 + \cdots a_d \equiv 1 \pmod 2$. The second condition implies that $P'(0) = a_1 \equiv 1 \pmod 2$, $P'(1) \equiv a_1 + a_3 + a_5 + \cdots \equiv 1 \pmod 2$. Now combining all this together we get $a_2 + a_3 + \cdots a_d \equiv 0 \pmod 2$ and $a_3 + a_5 + \cdots \equiv 0 \pmod 2$, hence $a_2 + a_4 + \cdots \equiv 0 \pmod 2$.

(5) As a bonus, we can use exactly the same proof to get exactly the same characterization of bijective modulo 2^r $(r = 1, 2, \ldots)$ mappings of the form $x \mapsto P(x) = a_0 \oplus a_1x \oplus \cdots \oplus a_dx^d \bmod 2^r$ since $u \oplus v$ is uniformly differentiable modulo 2 as bivariate function, and its derivative modulo 2 is exactly the same as the derivative of $u + v$, and besides, $u \oplus v \equiv u + v \pmod 2$.

Note that in general theorems 6.4 and 6.6 could be applied to a class of functions that is narrower than the class of all compatible functions. However, it turns out that for $p = 2$ this is not the case. Namely, the following proposition holds:

Proposition 6.8. ([3, Corollary 4.6], [4, Corollary 4.4]) *If a compatible function* $g \colon \mathbb{Z}_2 \to \mathbb{Z}_2$ *preserves measure then it is uniformly differentiable modulo* 2*, and its derivative modulo* 2 *is always* 1 *modulo* 2*.*

The above results are good to verify whether a given function preserves measure or is ergodic. However, we need more tools to construct measure-preserving, (respectively, ergodic) mappings in explicit form.

6.3. **Mahler's series.** We already have mentioned that uniformly continuous functions defined on (and valuated in) $\mathbb{Z}_p$ could be uniquely represented as Mahler's interpolation series (4.2.2). So, it is natural to express conditions of measure-preservation or ergodicity in terms of coefficients of these series.

Theorem 6.9 ([3, 4, 5])**.** *For* $p = 2$ *a function* $f \colon \mathbb{Z}_p \to \mathbb{Z}_p$ *is compatible and measure-preserving if and only if it could be represented as*

$$f(x) = c_0 + x + \sum_{i=1}^{\infty} c_i\, p^{\lfloor \log_p i \rfloor + 1} \binom{x}{i} \qquad (x \in \mathbb{Z}_p);$$

The function f *is compatible and ergodic if and only if it could be represented as*

$$f(x) = 1 + x + \sum_{i=1}^{\infty} c_i p^{\lfloor \log_p (i+1) \rfloor + 1} \binom{x}{i} \qquad (x \in \mathbb{Z}_p),$$

where $c_0, c_1, c_2 \ldots \in \mathbb{Z}_p$*. For* $p \neq 2$ *these conditions remain sufficient, and not necessary.*

Thus, in view of theorem 6.9 one can choose a state transition function to be a polynomial with rational (not necessarily integer) key-dependent coefficients setting $c_i = 0$ for all but finite number of i. Note that to determine whether a given polynomial f with rational (and not necessarily integer) coefficients is integer valued (that is, maps $\mathbb{Z}_p$ into itself), compatible and ergodic, it is sufficient to determine whether it induces a cycle on $O(\deg f)$ integral points. To be more exact, the following proposition holds.

Proposition 6.10 ([5]). *A polynomial $f(x)$ with rational, and not necessarily integer coefficients, is integer valued, compatible, and ergodic (resp., measure preserving) if and only if*

$$z \mapsto f(z) \bmod p^{\lfloor \log_p(\deg f)\rfloor+3},$$

where z runs through $0, 1, \ldots, p^{\lfloor \log_p(\deg f)\rfloor+3}-1$, is compatible and transitive (resp., bijective) mapping of the residue ring $\mathbb{Z}/p^{\lfloor \log_p(\deg f)\rfloor+3}$ onto itself.

Theorem 6.9 enables one to use exponentiation in design of generators that are transitive modulo 2^n for all $n = 1, 2, 3, \ldots$.

Example 6.11. For any odd $a = 1 + 2m$ a function $f(x) = ax + a^x$ defines a transitive modulo 2^n generator $x_{i+1} = f(x_i) \bmod 2^n$.

Indeed, in view of 6.9 the function f defines a compatible and ergodic mapping of $\mathbb{Z}_2$ onto $\mathbb{Z}_2$ since $f(x) = (1+2m)x + (1+2m)^x = x + 2mx + \sum_{i=0}^{\infty} m^i 2^i \binom{x}{i} = 1 + x + 4m\binom{x}{1} + \sum_{i=2}^{\infty} m^i 2^i \binom{x}{i}$ and $i \geq \lfloor \log_2(i+1)\rfloor + 1$ for all $i = 2, 3, 4, \ldots$.

Such a generator could be of practical value since it uses not more than $n+1$ multiplications modulo 2^n of n-bit numbers; of course, one should use calls to the table $a^{2^j} \bmod 2^n$, $j = 1, 2, 3, \ldots, n-1$. The latter table must be precomputed, corresponding calculations involve $n-1$ multiplications modulo 2^n. Obviously, one can use m as a long-term key, with the initial state x_0 being a short-term key, i.e., one changes m from time to time, but uses new x_0 for each new message. Obviously, without a properly chosen output function such a generator is not secure. The choice of output function in more details is discussed further.

Note. A similar argument shows that for every prime p and every $a \equiv 1 \pmod p$ the function $f(x) = ax + a^x$ defines a compatible and ergodic mapping of $\mathbb{Z}_p$ onto itself.

For polynomials with (rational or p-adic) integer coefficients theorem 6.9 may be restated in the following form.

Proposition 6.12 ([4, 3]). *Represent a polynomial $f(x) \in \mathbb{Z}_2[x]$ in a basis of descending factorial powers*

$$x^{\underline{0}} = 1,\ x^{\underline{1}} = x,\ x^{\underline{2}} = x(x-1), \ldots,\ x^{\underline{i}} = x(x-1)\cdots(x-i+1), \ldots,$$

i.e., let

$$f(x) = \sum_{i=0}^{d} c_i \cdot x^{\underline{i}}$$

for $c_0, c_1, \ldots, c_d \in \mathbb{Z}_2$. Then the polynomial f induces an ergodic (and, obviously, a compatible) *mapping of $\mathbb{Z}_2$ onto itself iff its coefficients c_0, c_1, c_2, c_3*

satisfy the following congruences:

$$c_0 \equiv 1 \pmod 2, \quad c_1 \equiv 1 \pmod 4, \quad c_2 \equiv 0 \pmod 2, \quad c_3 \equiv 0 \pmod 4.$$

The polynomial f induces a measure preserving mapping iff

$$c_1 \equiv 1 \pmod 2, \quad c_2 \equiv 0 \pmod 2, \quad c_3 \equiv 0 \pmod 2.$$

Thus, to provide ergodicity of the polynomial mapping f it is necessary and sufficient to hold fixed 6 bits only, while the other bits of coefficients of f may vary (e.g., may be key-dependent). This guarantees transitivity of the state transition function $z \mapsto f(z) \bmod 2^n$ for each n, and hence, uniform distribution of the output sequence.

Proposition 6.12 implies that the polynomial $f(x) \in \mathbb{Z}[x]$ is ergodic (resp., measure preserving) iff it is transitive modulo 8 (resp., iff it is bijective modulo 4). A corresponding assertion holds in general case, for arbitrary prime p.

Theorem 6.13 ([18, 5])**.** *A polynomial $f(x) \in \mathbb{Z}_p[x]$ induces an ergodic mapping of $\mathbb{Z}_p$ onto itself iff it is transitive modulo p^2 for $p \neq 2, 3$, or modulo p^3, for $p = 2, 3$. The polynomial $f(x) \in \mathbb{Z}_p[x]$ induces a measure preserving mapping of $\mathbb{Z}_p$ onto itself iff it is bijective modulo p^2.*

Example 6.14. The mapping $x \mapsto f(x) \equiv x + 2x^2 \pmod{2^{32}}$ (which is used in RC6, see [22]) is bijective, since it is bijective modulo 4: $f(0) \equiv 0 \pmod 4$, $f(1) \equiv 3 \pmod 4$, $f(2) \equiv 2 \pmod 4$, $f(3) \equiv 1 \pmod 4$. Thus, the mapping $x \mapsto f(x) \equiv x + 2x^2 \pmod{2^n}$ is bijective for all $n = 1, 2, \ldots$.

Hence, with the use of the theorem 6.13 it is possible to obtain transitive modulo $q > 1$ mappings for arbitrary natural q: one can just take $f(z) = (1 + z + \hat{q}g(z)) \bmod q$, where $g(x) \in \mathbb{Z}[x]$ is an arbitrary polynomial, and $\hat{q}$ is a product of p^{s_p} for all prime factors p of q, where $s_2 = s_3 = 3$, and $s_p = 2$ for $p \neq 2, 3$. Again, the polynomial $g(x)$ may be chosen, roughly speaking, 'more or less at random', i.e., it may be key-dependent, but the output sequence will be uniformly distributed for any choice of $g(x)$. This assertion may be generalized either.

Proposition 6.15 ([5])**.** *Let p be a prime, and let $g(x)$ be an arbitrary composition of arithmetic operations and mappings listed in* (4.2.1)*. Then the mapping $z \mapsto 1 + z + p^2 g(z)$ $(z \in \mathbb{Z}_p)$ is ergodic.*

In fact, both propositions 6.12, 6.15 and theorem 6.13 are special cases of the following general

Theorem 6.16 ([5])**.** *Let $\mathcal{B}_p$ be a class of all functions defined by series of a form $f(x) = \sum_{i=0}^{\infty} c_i \cdot x^{\underline{i}}$, where $c_0, c_1, \ldots$ are p-adic integers, and $x^{\underline{i}}$ $(i = 0, 1, 2, \ldots)$ are descending factorial powers* (see 6.12)*. Then the function $f \in \mathcal{B}_p$ preserves measure iff it is bijective modulo p^2; f is ergodic iff it is transitive modulo p^2 (for $p \neq 2, 3$), or modulo p^3 (for $p \in \{2, 3\}$).*

Note. As it was shown in [5], the class $\mathcal{B}_p$ contains all polynomial functions over $\mathbb{Z}_p$, as well as analytic (e.g., rational, entire) functions that are convergent everywhere on $\mathbb{Z}_p$. [10] As a matter of fact, every mapping that is a composition of arithmetic operators (addition, subtraction, multiplication, and

[10] More information about this class could be found in [1]

operators listed in (4.2.1)) belong to $\mathcal{B}_p$; thus, every such mapping modulo p^n could be induced by a polynomial with rational integer coefficients (see the end of Section 4 in [5]). For instance, the mapping $x \mapsto (3x+3^x) \bmod 2^n$ (which is transitive modulo 2^n, see 6.11) could be induced by a polynomial $1+x+4\binom{x}{1}+\sum_{i=2}^{n-1} 2^i\binom{x}{i} = 1+5x+\sum_{i=2}^{n-1} \frac{2^i}{i!}\cdot x^{\underline{i}}$ — just note that $c_i = \frac{2^i}{i!}$ are 2-adic integers since the exponent of maximal power of 2 that is a factor of $i!$ is exactly $i - \operatorname{wt}_2 i$, where $\operatorname{wt}_2 i$ is a number of 1's in the base-2 expansion of i (see e.g. [17, Chapter 1, Section 2, Exercise 12]); thus $\|c_i\|_2 = 2^{-\operatorname{wt}_2 i} \le 1$, i.e. $c_i \in \mathbb{Z}_2$ and so $c_i \bmod 2^n \in \mathbb{Z}$.

Theorem 6.16 implies that, for instance, the state transition function $f(z) = (1+z+\zeta(q)^2(1+\zeta(q)u(z))^{v(z)}) \bmod q$ is transitive modulo q for each natural $q > 1$ and arbitrary polynomials $u(x), v(x) \in \mathbb{Z}[x]$, where $\zeta(q)$ is a product of all prime factors of q. So the one can choose as a state transition function not only polynomial functions, but also rational functions, as well as analytic ones. It should be mentioned, however, that this is merely a form the function is represented (which could be suitable for some cases and unsuitable for the others), yet, for a given q, all the functions of this type may also be represented as polynomials over $\mathbb{Z}$ (see [5, Proposition 4.4; resp., Proposition 4.10 in the preprint]). For instance, certain generators of inversive kind (i.e., those using taking the inverse modulo 2^n) could be considered in such manner.

Example 6.17. For $f(x) = -\frac{1}{2x+1} - x$ a generator $x_{i+1} = f(x_i) \bmod 2^n$ is transitive. Indeed, the function $f(x) = (-1+2x-4x^2+8x^3-\cdots) - x = -1+x-4x^2+8(\cdots)$ is analytic and defined everywhere on $\mathbb{Z}_2$; thus $f \in \mathcal{B}_p$. Now the conclusion follows in view of 6.16 since by direct calculations it could be easily verified that the function $f(x) \equiv -1+x-4x^2 \pmod 8$ is transitive modulo 8. Note that modulo 2^n the mapping $x \mapsto f(x) \bmod 2^n$ could be induced by a polynomial $-1+x-4x^2+8x^3+\cdots+(-1)^n 2^{n-1}x^{n-1}$.

6.4. **Explicit expressions.** It turns out that there is an easy way to construct a measure preserving or ergodic mapping out of an arbitrary compatible mapping, i.e., out of an arbitrary composition of both arithmetic (including (4.2.1)) and bitwise logical operators.

Theorem 6.18 ([5])**.** *Let Δ be a difference operator, i.e., $\Delta g(x) = g(x+1) - g(x)$ by the definition. Let, further, p be a prime, let c be a coprime with p, $\gcd(c,p) = 1$, and let $g\colon \mathbb{Z}_p \to \mathbb{Z}_p$ be a compatible mapping. Then the mapping $z \mapsto c + z + p\Delta g(z)$ $(z \in \mathbb{Z}_p)$ is ergodic, and the mapping $z \mapsto d + cx + pg(x)$, preserves measure for arbitrary d.*

Moreover, if $p = 2$, then the converse also holds: Each compatible and ergodic (respectively each compatible and measure preserving) mapping $z \mapsto f(z)$ $(z \in \mathbb{Z}_2)$ could be represented as $f(x) = 1 + x + 2\Delta g(x)$ (respectively as $f(x) = d + x + 2g(x)$) for suitable $d \in \mathbb{Z}_2$ and compatible $g\colon \mathbb{Z}_2 \to \mathbb{Z}_2$.

Note. The case $p = 2$ is the only case the converse of the first assertion of theorem 6.18 holds.

Proof. To start with, by induction on l we show that g is bijective modulo p^l for all $l = 1, 2, 3, \ldots$. The assumption is obviously true for $l = 1$.

Assume it is true for $l = 1, 2, \dots, k-1$. Prove that it holds for $l = k$ either. Let $g(a) \equiv g(b) \pmod{p^k}$ for some p-adic integers a, b. Then $a \equiv b \pmod{p^{k-1}}$ by the induction hypothesis. Hence $pv(a) \equiv pv(b) \pmod{p^k}$ since v is compatible. Further, the congruence $g(a) \equiv g(b) \pmod{p^k}$ implies that $ca + pv(a) \equiv cb + pv(b) \pmod{p^k}$, and consequently, $ca \equiv cb \pmod{p^k}$. Since $c \not\equiv 0 \pmod p$, the latter congruence implies that $a \equiv b \pmod{p^k}$, proving the first assertion of the lemma.

To prove the rest part of the first assertion we note that the just proven claim implies that h preserves measure. To prove the transitivity of h modulo p^k for all $k = 1, 2, 3, \dots$ we apply induction on k once again.

It is obvious that h is transitive modulo p. Assume that h is transitive modulo p^{k-1}. Then, since h induces a permutation on the residue ring $\mathbb{Z}/p^k\mathbb{Z}$ and since h is a compatible function, we conclude that the length of each cycle of this permutation must be a multiple of p^{k-1}. Thus, to prove this permutation is single cycle it suffices to prove that the function

$$h^{p^{k-1}}(x) = \underbrace{h(h \dots (h}_{p^{k-1}}(x)) \dots)$$

induces a single cycle permutation on the ideal $p^{k-1}\mathbb{Z}$, generated by the element p^{k-1} of the ring $\mathbb{Z}/p^k\mathbb{Z}$. In other words, it is sufficient to demonstrate that the function $\frac{1}{p^{k-1}} h^{p^{k-1}}(p^{k-1}x)$ is transitive modulo p.

Applying obvious direct calculations, we successively obtain that

$$\begin{gathered} h^1(x) = c + x + pv(x+1) - pv(x), \\ \dots \quad \dots \quad \dots \\ h^j(x) = h(h^{j-1}(x)) = cj + h^{j-1}(x) + pv(h^{j-1}(x) + 1) - pv(h^{j-1}(x)) = \\ cj + x + p\sum_{i=0}^{j-1} v(h^i(x) + 1) - p\sum_{i=0}^{j-1} v(h^i(x)), \end{gathered}$$

and henceforth. We recall that $h^0(x) = x$ by the definition. So,

$$h^{p^{k-1}}(x) = cp^{k-1} + x + p\sum_{i=0}^{p^{k-1}-1} v(h^i(x)+1) - p\sum_{i=0}^{p^{k-1}-1} v(h^i(x)). \tag{6.18.1}$$

Since h is transitive modulo p^{k-1} and compatible, we get now that

$$\sum_{i=0}^{p^{k-1}-1} v(h^i(x)+1) \equiv \sum_{i=0}^{p^{k-1}-1} v(h^i(x)) \equiv \sum_{z=0}^{p^{k-1}-1} v(z) \pmod{p^{k-1}},$$

and (6.18.1) implies then $h^{p^{k-1}}(x) \equiv cp^{k-1} + x \pmod{p^k}$. But $c \not\equiv 0 \pmod p$, so we conclude that the function $cp^{k-1} + x$ induces on the ideal $p^{k-1}\mathbb{Z}$ a single cycle permutation, thus proving the first assertion of the theorem.

To prove the second assertion, note that as g is compatible, its Mahler's interpolation series are of the form of Theorem 4.3; noe note that $\Delta\binom{x}{i} = \binom{x}{i-1}$ and apply Theorem 6.9. $\square$

Example 6.19. Theorem 6.18 immediately implies Theorem 2 of [15]: For any composition f of primitive functions, the mapping $x \mapsto x + 2f(x) \pmod{2^n}$ is invertible — just note that a composition of primitive functions is compatible (see [15] for the definition of primitive functions). □

Theorem 6.18 is maybe one of the most important tools in design of pseudorandom generators such that both their state transition functions and output functions are key-dependent. The corresponding schemes are rather flexible: In fact, one may use nearly arbitrary composition of arithmetic and logical operators to produce a strictly uniformly distributed sequence: Both for $g(x) = x \,\mathsf{XOR}(2x+1)$ and for

$$g(x) = \left(1 + 2\frac{x \,\mathsf{AND}\, x^2 + x^3 \,\mathsf{OR}\, x^4}{3 + 4(5+6x^5)^{x^6 \,\mathsf{XOR}\, x^7}}\right)^{7+\frac{8x^8}{9+10x^9}}$$

a sequence $\{x_i\}$ defined by recurrence relation $x_{i+1} = (1 + x_i + 2(g(x_i+1) - g(x_i))) \bmod 2^n$ is strictly uniformly distributed in $\mathbb{Z}/2^n\mathbb{Z}$ for each $n = 1, 2, 3\ldots$, i.e., the sequence $\{x_i\}$ is purely periodic with *period length exactly* 2^n, and *each* element of $\{0, 1, \ldots, 2^n - 1\}$ occurs at the period *exactly once*. We will demonstrate further that a designer could vary the function g in a very wide scope without worsening prescribed values of some important indicators of security. In fact, choosing the proper arithmetic and bitwise logical operators the designer is restricted only by desirable performance, since any compatible ergodic mapping could be produced in this way:

Corollary 6.20. *Let $p = 2$, and let f be a compatible and ergodic mapping of $\mathbb{Z}_2$ onto itself. Then for each $n = 1, 2, \ldots$ the state transition function f mod 2^n could be represented as a finite composition of arithmetic and bitwise logical operators.*

Proof. In view of proposition 6.18 it is sufficient to prove that for arbitrary compatible g the function $\bar{g} = g \bmod 2^n$ could be represented as a finite composition of operators mentioned in the statement. In view of Definition 2.1,one could represent $\bar{g}$ as

$$\bar{g}(x) = \gamma_0(\chi_0) + 2\gamma_1(\chi_0, \chi_1) + \cdots + 2^{n-1}\gamma_{n-1}(\chi_0, \ldots, \chi_{n-1}),$$

where $\gamma_i = \delta_i(\bar{g})$, $\chi_i = \delta_i(x)$, $i = 0, 1, \ldots, n-1$. Since each $\gamma_i(\chi_0, \ldots, \chi_i)$ is a Boolean function in Boolean variables $\chi_0, \ldots, \chi_i$, it could be expressed via finite number of XORs and ANDs of these variables $\chi_0, \ldots, \chi_i$. Yet each variable χ_j could be expressed as $\chi_j = \delta_j(x) = x \,\mathsf{AND}(2^j)$, and the conclusion follows. □

6.5. **Using Boolean representations.** As we just have seen, in case $p = 2$ we have two equivalent descriptions of the class of all compatible ergodic mappings, namely, theorems 6.9 and 6.18. They enable one to express *any* compatible and transitive modulo 2^n state transition function either as a polynomial of special kind over a field $\mathbb{Q}$ of rational numbers, or as a special composition of arithmetic and bitwise logical operations. Both these representations are suitable for programming, since they involve only standard machine instructions. However, we need one more representation, in a Boolean form, which we have already used in the definition of T-function

(see Definition 2.1). Despite this representation is not very convenient for programming, it could be used to prove the ergodicity of some simple mappings, see e.g. 6.22 below. The following theorem is just a restatement in our terms of a known (at least 30 years old) result from the theory of Boolean functions, the so-called bijectivity/transitivity criterion for triangle Boolean mappings. However, the latter result is a mathematical folklore, and thus it is somewhat difficult to attribute it.

Recall that the *algebraic normal form*, ANF, of the Boolean function $\psi_j(\chi_0, \dots, \chi_j)$ is the representation of this function via $\oplus$ (addition modulo 2, that is, logical 'exclusive or') and $\cdot$ (multiplication modulo 2, that is, logical 'and', or conjunction). In other words, the ANF of the Boolean function ψ is its representation in the form

$$\psi(\chi_0, \dots, \chi_j) = \beta \oplus \beta_0\chi_0 \oplus \beta_1\chi_1 \oplus \dots \oplus \beta_{0,1}\chi_0\chi_1 \oplus \dots,$$

where $\beta, \beta_0, \dots \in \{0, 1\}$. The ANF is sometimes called a *Boolean polynomial*.

Recall that the *weight* of the Boolean function ψ_j in $(j+1)$ variables is the number of $(j+1)$-bit words that *satisfy* ψ_j; that is, weight is the cardinality of the truth set of ψ_j.

Theorem 6.21. *A mapping $T\colon \mathbb{Z}_2 \to \mathbb{Z}_2$ is compatible and measure preserving iff for each $i = 0, 1, \dots$ the ANF of the Boolean function $\tau_i^T = \delta_i(T)$ in Boolean variables $\chi_0, \dots, \chi_i$ is*

$$\tau_i^T(\chi_0, \dots, \chi_i) = \chi_i \oplus \varphi_i^T(\chi_0, \dots, \chi_{i-1}),$$

where φ_i^T is an ANF. The mapping T is compatible and ergodic iff, additionally, the Boolean function φ_i^T is of odd weight, that is, takes value 1 exactly at the odd number of points $(\varepsilon_0, \dots, \varepsilon_{i-1})$, where $\varepsilon_j \in \{0, 1\}$ for $j = 0, 1, \dots, i-1$. The latter takes place if and only if $\varphi_0^T = 1$, and the degree of the ANF φ_i^T for $i \geq 1$ is exactly i, that is, φ_i^T contains a monomial $\chi_0 \cdots \chi_{i-1}$.

Proof. Represent the value of the function T at the 2-adic integer point $x = \chi_0 + \chi_1 \cdot 2 + \chi_2 \cdot 2^2 + \cdots$ as a 2-adic integer:

$$T(\chi_0 + \chi_1 \cdot 2 + \chi_2 \cdot 2^2 + \cdots) = \sum_{i=0}^{\infty} \delta_i(x) \cdot 2^i.$$

The function T is compatible (that is, a T-function) if and only if $\delta_i(x)$ does not depend on $\chi_{i+1}, \chi_{i+2}, \dots$ for every $i = 0, 1, 2, \dots$, see Definition 2.1. Thus, each $\delta_i(x)$ is a Boolean function τ_i^T in Boolean variables $\chi_0, \chi_1, \dots, \chi_i$. Re-write the ANF of the function τ_i^T in the following form:

$$\tau_i^T(\chi_0, \dots, \chi_i) = \chi_i \cdot \psi_i^T(\chi_0, \dots, \chi_{i-1}) \oplus \varphi_i^T(\chi_0, \dots, \chi_{i-1}),$$

where both $\psi_i^T(\chi_0, \dots, \chi_{i-1})$ and $\varphi_i^T(\chi_0, \dots, \chi_{i-1})$ are Boolean functions in Boolean variables $\chi_0, \dots, \chi_{i-1}$.

Obviously, whenever all $\psi_i^T(\chi_0, \dots, \chi_{i-1})$ are identically 1, the function is measure-preserving since it is bijective modulo 2^{k+1} for each $k = 0, 1, 2, \dots$: To find a co-image of the mapping $T \bmod 2^k$ one must solve a system of

Boolean equations

$$\begin{cases} \chi_0 + \varphi_0^T & = \alpha_0, \\ \chi_1 + \varphi_1^T(\chi_1) & = \alpha_1, \\ \dots\dots\dots\dots\dots\dots\dots & \dots\dots \\ \chi_k + \varphi_k^T(\chi_0, \dots, \chi_{k-1}) & = \alpha_k, \end{cases}$$

which has a unique solution given any $\alpha_0, \dots, \alpha_k \in \{0, 1\}$.

Conversely, in let i be the smallest number such that $\psi_i(\chi_0, \dots, \chi_{i-1}) = 0$ for a certain set $\chi_0, \dots, \chi_{i-1}$ of zeros and ones. Then

$$T(\chi_0+\chi_1\cdot 2+\cdots\chi_2\cdot 2^{i-1}+0\cdot 2^i) \equiv T(\chi_0+\chi_1\cdot 2+\cdots\chi_2\cdot 2^{i-1}+1\cdot 2^i) \pmod{2^{i+1}}.$$

Thus, T can not be measure-preserving in view of Theorem 5.2.

Further, to prove the ergodicity part of the statement we note that T is transitive modulo 2 if and only if $\tau_0^T(\chi_0) = \chi_0 \oplus 1$. In case T is transitive modulo 2^k,

$$\delta_i(T^{2^k})(x) = \begin{cases} \chi_i, & \text{if } i < k; \\ \chi_k \oplus \sigma, & \text{if } i = k, \end{cases}$$

where σ is a sum modulo 2 of all values of the Boolean function φ_k^T at all points of $\mathbb{B}^k$; that is, σ is the weight modulo 2 of the function φ_k^T. Clearly, to provide transitivity of the function T modulo 2^{k+1}, (cf. Theorem 5.2 must be $\sigma = 1$. That is, weight of the function φ_k^T must be odd.

The rest of the statement of the theorem is a well-known result in the theory of Boolean functions; the proof is left to a reader. □

Note. The bit-slice techniques of Klimov and Shamir, which they introduced in 2002 in [15] is just a re-statement of the above stated folklore theorem 6.21.

This is how Theorem 6.21 works:

Example 6.22. With the use of 6.21 it is possible to give another proof of the main result of [15], namely, of Theorem 3: *The mapping $f(x) = x + (x^2 \vee C)$ over n-bit words is invertible if and only if the least significant bit of C is 1. For $n \geq 3$ it is a permutation with a single cycle if and only if both the least significant bit and the third least significant bit of C are* 1.

Proof of theorem 3 of [15]. Recall that for $x \in \mathbb{Z}_2$ and $i = 0, 1, 2, \dots$ we denote $\chi_i = \delta_i(x) \in \{0, 1\}$; also we denote $c_i = \delta_i(C)$. We will calculate $\delta_i(x + (x^2 \vee C))$ as an ANF in Boolean variables $\chi_0, \chi_1, \dots$ and we start with the following easy claims:

- $\delta_0(x^2) = \chi_0$, $\delta_1(x^2) = 0$, $\delta_2(x^2) = \chi_0\chi_1 \oplus \chi_1$,
- $\delta_n(x^2) = \chi_{n-1}\chi_0 \oplus \psi_n(\chi_0, \dots, \chi_{n-2})$ for all $n \geq 3$, where ψ_n is a Boolean function in $n-1$ Boolean variables $\chi_0, \dots, \chi_{n-2}$.

The first of these claims could be easily verified by direct calculations. To prove the second one represent $x = \bar{x}_{n-1} + 2^{n-1}s_{n-1}$ (where we recall $\bar{x}_{n-1} = x \bmod 2^{n-1}$) and calculate $x^2 = (\bar{x}_{n-1} + 2^{n-1}s_{n-1})^2 = \bar{x}_{n-1}^2 + 2^n s_{n-1}\bar{x}_{n-1} + 2^{2n-2}s_{n-1}^2 = \bar{x}_{n-1}^2 + 2^n\chi_{n-1}\chi_0 \pmod{2^{n+1}}$ for $n \geq 3$ and note that $\bar{x}_{n-1}^2$ depends only on $\chi_0, \dots, \chi_{n-2}$.

This gives

(1) $\delta_0(x^2 \vee C) = \chi_0 \oplus c_0 \oplus \chi_0 c_0$
(2) $\delta_1(x^2 \vee C) = c_1$
(3) $\delta_2(x^2 \vee C) = \chi_0\chi_1 \oplus \chi_1 \oplus c_2 \oplus c_2\chi_1 \oplus c_2\chi_0\chi_1$
(4) $\delta_n(x^2 \vee C) = \chi_{n-1}\chi_0 \oplus \psi_n \oplus c_n \oplus c_n\chi_{n-1}\chi_0 \oplus c_n\psi_n$ for $n \geq 3$

From here it follows that if $n \geq 3$, then $\delta_n(x^2 \vee C) = \lambda_n(\chi_0, \dots, \chi_{n-1})$, and $\deg \lambda_n \leq n-1$, since ψ_n depends only on, may be, $\chi_0, \dots, \chi_{n-2}$.

Now successively calculate $\gamma_n = \delta_n(x + (x^2 \vee C))$ for $n = 0, 1, 2, \dots$. We have $\delta_0(x + (x^2 \vee C)) = c_0 \oplus \chi_0 c_0$ so necessarily $c_0 = 1$ since otherwise f is not bijective modulo 2. Proceeding further with $c_0 = 1$ we obtain $\delta_1(x + (x^2 \vee C)) = c_1 \oplus \chi_0 \oplus \chi_1$, since χ_1 is a carry. Then $\delta_2(x + (x^2 \vee C)) = (c_1\chi_0 \oplus c_1\chi_1 \oplus \chi_0\chi_1) \oplus (\chi_0\chi_1 \oplus \chi_1 \oplus c_2 \oplus c_2\chi_1 \oplus c_2\chi_0\chi_1) \oplus \chi_2 = c_1\chi_0 \oplus c_1\chi_1 \oplus \chi_1 \oplus c_2 \oplus c_2\chi_1 \oplus c_2\chi_0\chi_1 \oplus \chi_2$, here $c_1\chi_0 \oplus c_1\chi_1 \oplus \chi_0\chi_1$ is a carry. From here in view of 6.21 we immediately have $c_2 = 1$ since otherwise f is not transitive modulo 8. Now for $n \geq 3$ one has $\gamma_n = \alpha_n + \lambda_n \oplus \chi_n$, where α_n is a carry, and $\alpha_{n+1} = \alpha_n\lambda_n \oplus \alpha_n\chi_n \oplus \lambda_n\chi_n$. But if $c_2 = 1$ then $\deg \alpha_3 = \deg(\mu\nu \oplus \chi_2\mu \oplus \chi_2\nu) = 3$, where $\mu = c_1\chi_0 \oplus c_1\chi_1 \oplus \chi_0\chi_1$, $\nu = (\chi_0\chi_1 \oplus \chi_1 \oplus c_2 \oplus c_2\chi_1 \oplus c_2\chi_0\chi_1) = 0$. This implies inductively in view of (4) above that $\deg \alpha_{n+1} = n+1$ and that $\gamma_{n+1} = \chi_{n+1} \oplus \xi_{n+1}(\chi_0, \dots, \chi_n)$, $\deg \xi_{n+1} = n+1$. So the conditions of 6.21 are satisfied, thus finishing the proof of theorem 3 of [15]. □

There are some more applications of Theorem 6.21.

Proposition 6.23. *Let $F\colon \mathbb{Z}_2^{n+1} \to \mathbb{Z}_2$ be a compatible mapping such that for all $z_1, \dots, z_n \in \mathbb{Z}_2$ the mapping $F(x, z_1, \dots, z_n)\colon \mathbb{Z}_2 \to \mathbb{Z}_2$ is measure preserving. Then $F(f(x), 2g_1(x), \dots, 2g_n(x))$ preserves measure for all compatible $g_1, \dots, g_n\colon \mathbb{Z}_2 \to \mathbb{Z}_2$ and all compatible and measure preserving $f\colon \mathbb{Z}_2 \to \mathbb{Z}_2$. Moreover, if f is ergodic then $f(x + 4g(x))$, $f(x \oplus (4g(x)))$, $f(x) + 4g(x)$, and $f(x) \oplus (4g(x))$ are ergodic for any compatible $g\colon \mathbb{Z}_2 \to \mathbb{Z}_2$* (here $\oplus$ stands for XOR).

Proof. Try to prove this yourself! □

Example 6.24. With the use of 6.23 it is possible to construct very fast generators $x_{i+1} = f(x_i) \bmod 2^n$ that are transitive modulo 2^n. For instance, take

$$f(x) = (\dots((((x + c_0) \oplus d_0) + c_1) \oplus d_1) + \dots + c_m) \oplus d_m,$$

where $c_0 \equiv 1 \pmod 2$, and the rest of c_i, d_i are 0 modulo 4. By the way, this generator, looking somewhat 'linear', is as a rule rather 'nonlinear': the corresponding polynomial over $\mathbb{Q}$ is of high degree. The general case of these functions f (for arbitrary c_i, d_i) was studied by the author's student Ludmila Kotomina: She proved that such a function is ergodic iff it is transitive modulo 4.

Yet another application of Theorem 6.21 are multivariate single cycle T-functions. We already know that there are no such functions among uniformly differentiable modulo 2 functions, see Theorem 6.3. However, the non-differentiable modulo 2 multivariate ergodic functions on $\mathbb{Z}_2$ exist.

In 2004 Klimov and Shamir introduced a multivariate T-function H with a single cycle property. The m-variate mapping

$$H\colon (\vec{x}_0, \vec{x}_1, \dots, \vec{x}_{m-1}) \mapsto (h_0, h_1, \dots, h_{m-1})$$

over n-bit words $\overrightarrow{x}_0, \overrightarrow{x}_1, \dots, \overrightarrow{x}_{m-1}$, defined by

$$h_s = \overrightarrow{x}_s \oplus ((h(\overrightarrow{x}_0 \wedge \dots \wedge \overrightarrow{x}_{m-1}) \oplus (\overrightarrow{x}_0 \wedge \dots \wedge \overrightarrow{x}_{m-1})) \wedge \overrightarrow{x}_0 \wedge \dots \wedge \overrightarrow{x}_{s-1},$$

$s = 0, 1, \dots, m-1$, has a single cycle property whenever h is a univariate T-function with a single cycle property. Here $\wedge$ stands for AND, bitwise logical 'and' (a conjunction). We assume that a bitwise conjunction over an empty set of indices is a string of all 1's.

Actually, this is just a trick: The m-variate mapping H on n-bit words is a multivariate representation of a univariate T-function over mn-bit words. Indeed, given a univariate T-function F,

$$x = (\dots, \chi_2, \chi_1, \chi_0) \overset{F}{\mapsto} (\dots; \psi_2(\chi_0, \chi_1, \chi_2); \psi_1(\chi_0, \chi_1); \psi_0(\chi_0)),$$

arrange this mapping in columns of height m, this way:

$$\begin{array}{lllclll}
\dots \chi_{2m} & \chi_m & \chi_0 & \overset{f_0}{\mapsto} & \dots \psi_{2m}(x) & \psi_m(x) & \psi_0(x) \\
\dots \chi_{2m+1} & \chi_{m+1} & \chi_1 & \overset{f_1}{\mapsto} & \dots \psi_{2m+1}(x) & \psi_{m+1}(x) & \psi_1(x) \\
\dots & \dots & \dots & \dots & & & \\
\dots \chi_{3m-1} & \chi_{2m-1} & \chi_{m-1} & \overset{f_{m-1}}{\mapsto} & \dots \psi_{3m-1}(x) & \psi_{2m-1}(x) & \psi_{m-1}(x)
\end{array}$$

Now just assume the left-hand rows are new variables:

$$\overrightarrow{x}_j = (\dots, \chi_{2m+j}, \chi_{m+j}, \chi_j), \qquad (j = 0, 1, \dots, m-1).$$

Obviously, the m-variate mapping $\mathbf{F} = (f_0, f_1, \dots, f_{m-1})$ has a single cycle property iff a univariate mapping F has a single property.

Consider the simplest example: $F(x) = 1 + x$. We have

$$\delta_j(F(x)) \equiv \delta_j(x) + \prod_{s=0}^{j-1} \delta_s(x) \pmod 2$$

(we assume the product over the empty set is 1); then the m-variate representation $\mathbf{F} = (f_0, f_1, \dots, f_{m-1})$ of this mapping is

$$f_k(\overrightarrow{x}_0, \dots, \overrightarrow{x}_{m-1}) = \overrightarrow{x}_k \oplus \left(\left(\bigwedge_{s=0}^{k-1} \overrightarrow{x}_s \right) \wedge \left(\bigwedge_{r=0}^{m-1} ((\overrightarrow{x}_r + 1) \oplus \overrightarrow{x}_r) \right) \right) =$$

$$\overrightarrow{x}_k \oplus \left(\left(\bigwedge_{s=0}^{k-1} \overrightarrow{x}_s \right) \wedge \left(\left(\left(\bigwedge_{r=0}^{m-1} \overrightarrow{x}_r \right) + 1 \right) \oplus \left(\bigwedge_{r=0}^{m-1} \overrightarrow{x}_r \right) \right) \right).$$

With the use of this trick and with Theorem 6.21 the following multivariate ergodic T-functions could be constructed:

Proposition 6.25 ([7]). *Let $t, j \in \{0, 1, \dots, m-1\}$, let all $f_j^{(t)}$ (resp., $g_j^{(t)}$) be univariate ergodic (resp, measure-preserving) compatible mappings from*

$\mathbb{Z}_2$ onto $\mathbb{Z}_2$. Then the mapping $\mathbf{F}(\mathbf{x}) = (f_0(\mathbf{x}), \dots, f_{m-1}(\mathbf{x}))$

$$f_0(\mathbf{x}) = \overrightarrow{x}_0 \boxplus \left(\bigwedge_{r=0}^{m-1} (f_0^{(r)}(\overrightarrow{x}_r) \oplus \overrightarrow{x}_r) \right);$$

$$f_1(\mathbf{x}) = \overrightarrow{x}_1 \boxplus \left(g_1^{(0)}(\overrightarrow{x}_0) \wedge \left(\bigwedge_{r=0}^{m-1} (f_1^{(r)}(\overrightarrow{x}_r) \oplus \overrightarrow{x}_r) \right) \right);$$

....

$$f_{m-1}(\mathbf{x}) = \overrightarrow{x}_{m-1} \boxplus \left(\left(\bigwedge_{t=0}^{m-2} g_{m-1}^{(t)}(\overrightarrow{x}_t) \right) \wedge \left(\bigwedge_{r=0}^{m-1} (f_{m-1}^{(r)}(\overrightarrow{x}_r) \oplus \overrightarrow{x}_r) \right) \right),$$

where $\mathbf{x} = (\overrightarrow{x}_0, \dots, \overrightarrow{x}_{m-1})$, $\boxplus \in \{+, \oplus\}$, *is a compatible and ergodic mapping of* $\mathbb{Z}_2^m$ *onto* $\mathbb{Z}_2^m$.

7. Wreath products of PRNGs

In the preceding section we have developed some tools that enable us to construct algorithms based on standard instructions of an n-bit word processor that produce strictly uniformly distributed sequences of period length 2^n.

To judge whether these sequences could be of use for stream encryption we must study their properties that are crucial for stream ciphers. One of these properties is long period. But is the period of are sequences long enough? Not yet! In case $n = 32$, which is a standard for most contemporary processors, we obtain a period of length 2^{32}, which is too small to satisfy contemporary safety conditions: At least some 2^{80} is needed. Thus, we must make the period longer *leaving the sequence uniformly distributed*. In this section we consider corresponding techniques.

7.1. What is wreath product.

We start with a formal definition:

Definition 7.1. Given a mapping $U\colon Z \to Z$, and a set of mappings $\mathcal{V} = \{(V_z\colon X \to X)\colon z \in Z\}$, a *wreath product* (or, a *skew product* or, a *skew shift*) is a mapping

$$U \leftthreetimes \mathcal{V}\colon (z, x) \mapsto (U(z), V_z(x))$$

of the Cartesian product $Z \times X$ into itself.

In other words, the wreath product is a bivariate mapping where the first coordinate is a function of the variable z *only*, and the second coordinate is a bivariate function of z and x.

Most probably, you are already familiar with examples of wreath products; recall Feistel network: The mapping it is based on is $(z, x) \mapsto (z, z \oplus f(x))$, where $z, x \in \mathbb{B}^n$, $f\colon \mathbb{B}^n \to \mathbb{B}^n$, which is obviously a wreath product of $U(z) = z$ with $\mathcal{V} = \{V_z(x) = z \oplus f(x)\colon z \in \mathbb{B}^n\}$.

> *Obviously, the wreath product $U \leftthreetimes \mathcal{V}$ is bijective whenever both U and all V_z are bijective.*

Some terminology notes: In automata theory (and in algebra) they used to speak of wreath products, whereas in dynamical systems (and in ergodic

theory) theory they prefer the term skew product, or skew shift. Recall that ordinary PRNG corresponds to an *autonomous* dynamical system.

This is a *non-autonomous* dynamical system, which is a counterpart of a counter-dependent PRNG 2.0.1 in dynamics: A non-autonomous dynamical system is a dynamical system driven by another dynamical system, and skew products are used to combine two dynamical systems into a new one.

Note that a T-function is a composition of wreath products: Let F be a T-function,

$$(\chi_0, \chi_1, \chi_2, \ldots) \overset{F}{\mapsto} (\psi_0(\chi_0); \psi_1(\chi_0, \chi_1); \psi_2(\chi_0, \chi_1, \chi_2); \ldots),$$

then

$$\begin{array}{rcl} \chi_0 & \mapsto & \psi_0(\chi_0) \\ (\chi_0, \chi_1) & \mapsto & (\psi_0(\chi_0), \psi_1(\chi_0, \chi_1)) \\ ((\chi_0, \chi_1), \chi_2) & \mapsto & ((\psi_0(\chi_0), \psi_1(\chi_0, \chi_1)), \psi_2(\chi_0, \chi_1, \chi_2)) \\ \ldots\ldots\ldots\ldots & \ldots & \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \end{array}$$

Now we re-state the above definition for the case of wreath products of automata:

Definition 7.2. Let $\mathfrak{A}_j = \langle N, M, f_j, F_j \rangle$ be a family of automata with the same state set N and the same output alphabet M indexed by elements of a non-empty (possibly, countably infinite) set J (members of the family need not be necessarily pairwise distinct). Let $T\colon J \to J$ be an arbitrary mapping. A *wreath product* of the family $\{\mathfrak{A}_j\}$ of automata with respect to the mapping T is an automaton with the state set $N \times J$, state transition function $\breve{f}(j, z) = (f_j(z), T(j))$ and output function $\breve{F}(j, z) = F_j(z)$. We call f_j (resp., F_j) *clock* state update (resp., output) functions.

Obviously, the state transition function $\breve{f}(j, z) = (f_j(z), T(j))$ is a wreath product of a family of mappings $\{f_j\colon j \in J\}$ with respect to the mapping T

It worth notice here that if $J = \mathbb{N}_0$ and F_i does not depend on i, this construction gives us a number of examples of counter-dependent generators in the sense of [23, Definition 2.4], where the notion of a counter-dependent generator was originally introduced. However, we use this notion in a broader sense in comparison with that of [23]: In our counter-dependent generators not only the state transition function, but also the output function depends on i. Moreover, in [23] only a special case of counter-dependent generators is studied; namely, counter-assisted generators and their cascaded and two-step modifications. A state transition function of a counter-assisted generator is of the form $f_i(x) = i \star h(x)$, where $\star$ is a binary quasigroup operation (in particular, group operation, e.g., $+$ or XOR), and $h(x)$ does not depend on i. An output function of a counter-assisted generator does not depend on i either.

7.2. **Constructions.** In this subsection we introduce a method to construct counter dependent pseudorandom generators out of ergodic and measure-preserving mappings. The method guarantees that output sequences of these generators are always strictly uniformly distributed. Actually, all these constructions are wreath products of automata in the sense of 7.2; the following results give us conditions these automata should satisfy to produce a uniformly distributed output sequence. Our main technical tool is the following

theorem, which actually could be considered as a generalization of Theorem 6.21:

Theorem 7.3 ([6]). *Let $\mathcal{G} = g_0, \dots, g_{m-1}$ be a finite sequence of compatible measure preserving mappings of $\mathbb{Z}_2$ onto itself such that*

(1) *the sequence $\{(g_{i \bmod m}(0)) \bmod 2 \colon i = 0, 1, 2, \dots\}$ is purely periodic, its shortest period is of length m;*
(2) $\sum_{i=0}^{m-1} g_i(0) \equiv 1 \pmod 2$;
(3) $\sum_{j=0}^{m-1} \sum_{z=0}^{2^k-1} g_j(z) \equiv 2^k \pmod{2^{k+1}}$ *for all $k = 1, 2, \dots$.*

Then the recurrence sequence $\mathcal{Z}$ defined by the relation $x_{i+1} = g_{i \bmod m}(x_i)$ is strictly uniformly distributed modulo 2^n for all $n = 1, 2, \dots$: That is, modulo each 2^n the sequence $\mathcal{Z}$ is purely periodic, its shortest period is of length $2^n m$, and each element of $\mathbb{Z}/2^n\mathbb{Z}$ occurs at the period exactly m times.

Note. In view of 6.21 condition (3) of theorem 7.3 could be replaced by the equivalent condition

$$\sum_{j=0}^{m-1} \mathrm{Coef}_{0,\dots,k-1}(\varphi_k^j) \equiv 1 \pmod 2 \qquad (k = 1, 2, \dots),$$

where $\mathrm{Coef}_{0,\dots,k-1}(\varphi)$ is a coefficient of the monomial $\chi_0 \cdots \chi_{k-1}$ in ANF φ.

It turns out that the sequence $\mathcal{Z}$ of 7.3 is just the sequence $\mathcal{Y}$ of the following

Lemma 7.4 ([6]). *Let $c_0, \dots, c_{m-1}$ be a finite sequence of 2-adic integers, and let $g_0, \dots, g_{m-1}$ be a finite sequence of compatible mappings of $\mathbb{Z}_2$ onto itself such that*

(i) $g_j(x) \equiv x + c_j \pmod 2$ *for $j = 0, 1, \dots, m-1$,*
(ii) $\sum_{j=0}^{m-1} c_j \equiv 1 \pmod 2$,
(iii) *the sequence $\{c_{i \bmod m} \bmod 2 \colon i = 0, 1, 2, \dots\}$ is purely periodic, its shortest period is of length m,*
(iv) $\delta_k(g_j(z)) \equiv \zeta_k + \varphi_k^j(\zeta_0, \dots, \zeta_{k-1}) \pmod 2$*, $k = 1, 2, \dots$, where $\zeta_r = \delta_r(z)$, $r = 0, 1, 2, \dots$,*
(v) *for each $k = 1, 2, \dots$ an odd number of ANFs φ_k^j in Boolean variables $\zeta_0, \dots, \zeta_{k-1}$ are of odd weight.*

Then the recurrence sequence $\mathcal{Y} = \{x_i \in \mathbb{Z}_2\}$ defined by the relation $x_{i+1} = g_{i \bmod m}(x_i)$ is strictly uniformly distributed: It is purely periodic modulo 2^k for all $k = 1, 2, \dots$; its shortest period is of length $2^k m$; each element of $\mathbb{Z}/2^k\mathbb{Z}$ occurs at the period exactly m times. Moreover,

(1) *the sequence $\mathcal{D}_s = \{\delta_s(x_i) \colon i = 0, 1, 2, \dots\}$ is purely periodic; it has a period of length $2^{s+1}m$,*
(2) $\delta_s(x_{i+2^s m}) \equiv \delta_s(x_i) + 1 \pmod 2$ *for all $s = 0, 1, \dots, k-1$, $i = 0, 1, 2, \dots$,*
(3) *for each $t = 1, 2, \dots, k$ and each $r = 0, 1, 2, \dots$ the sequence*

$$x_r \bmod 2^t, x_{r+m} \bmod 2^t, x_{r+2m} \bmod 2^t, \dots$$

is purely periodic, its shortest period is of length 2^t, each element of $\mathbb{Z}/2^t\mathbb{Z}$ occurs at the period exactly once.

Note 7.5. Assuming $m = 1$ in 7.3 one obtains ergodicity criterion 6.21.

Corollary 7.6 ([6])**.** *Let a finite sequence of mappings $\{g_0, \dots, g_{m-1}\}$ of $\mathbb{Z}_2$ into itself satisfy conditions of theorem 7.3, and let $\{F_0, \dots, F_{m-1}\}$ be an arbitrary finite sequence of balanced (and not necessarily compatible) mappings of $\mathbb{Z}/2^n\mathbb{Z}$ $(n \ge 1)$ onto $\mathbb{Z}/2^k\mathbb{Z}$, $1 \le k \le n$. Then the sequence $\mathcal{F} = \{F_{i \bmod m}(x_i)\colon i = 0, 1, 2\dots\}$, where $x_{i+1} = g_{i \bmod m}(x_i) \bmod 2^n$, is strictly uniformly distributed over $\mathbb{Z}/2^k\mathbb{Z}$: It is purely periodic with a period of length $2^n m$, and each element of $\mathbb{Z}/2^k\mathbb{Z}$ occurs at the period exactly $2^{n-k}m$ times.*

Theorem 7.3 and lemma 7.4 together with corollary 7.6 enables one to construct a counter-dependent generator out of the following components:

- A sequence $c_0, \dots, c_{m-1}$ of integers, which we call a *control sequence*.
- A sequence $h_0, \dots, h_{m-1}$ of compatible mappings, which is used to form a sequence of clock state update functions g_i
- A sequence $H_0, \dots, H_{m-1}$ of compatible mappings to produce clock output functions F_i

Note that ergodic functions that are needed could be produced out of compatible ones with the use of 6.18 or 6.23. A control sequence could be produced by an external generator (which in turn could be a generator of the kind considered in this course), or it could be just a queue the state update and output functions are called from a look-up table. The functions h_i and/or H_i could be either precomputed to arrange that look-up table, or they could be produced on-the-fly in a form that is determined by a control sequence. This form may also look 'crazy', e.g.,

$$h_i(x) = (\cdots((u_0(\delta_0(c_i)) \bigcirc_{\delta_1(c_i),\delta_2(c_i)} u_1(\delta_3(c_i))) \bigcirc_{\delta_4(c_i),\delta_5(c_i)} u_2(\delta_6(c_i)))\cdots, \tag{7.6.1}$$

where $u_j(0) = x$, the variable, and $u_j(1)$ is a constant (which is determined by c_i, or is read from a precomputed look-up table, etc.), while (say) $\bigcirc_{0,0} = +$, an integer addition, $\bigcirc_{1,0} = \cdot$, an integer multiplication, $\bigcirc_{0,1} = \mathsf{XOR}$, $\bigcirc_{1,1} = \mathsf{AND}$. This is absolutely no matter what these h_i and H_i look like or how they are obtained, *the above stated results give a general method to combine all the data together to produce a uniformly distributed output sequence of a maximum period length.*

Examples 7.7 ([6]). A basic circuit illustrating these example wreath products is given at Figure 3.

(1) Let $c_0, \dots, c_{m-1}$ be an arbitrary sequence of length $m = 2^s$, and let $\hat{h}_0, \dots, \hat{h}_{m-1}$ be arbitrary compatible mappings. For $0 \le j \le m-1$ put $h_j(x) = 1 + x + 4 \cdot \hat{h}_j(x)$ and let $g_j(x) = c_j + h_j(x)$. These mappings g_j satisfy conditions of theorem 7.3 if and only if $\sum_{j=0}^{2^m-1} c_j \equiv 1 \pmod 2$.

(2) For $m > 1$ odd let $\{h_0, \dots, h_{m-1}\}$ be a finite sequence of compatible and ergodic mappings; let $c_0, \dots, c_{m-1}$ be a finite sequence of integers such that
   - $\sum_{j=0}^{m-1} c_j \equiv 0 \pmod 2$, and
   - the sequence $\{c_{i \bmod m} \bmod 2\colon i = 0, 1, 2, \dots\}$ is purely periodic with the shortest period of length m.

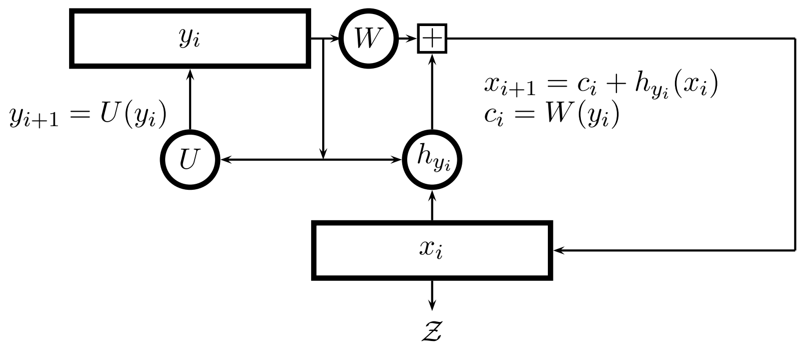

FIGURE 3. Wreath product basic circuit for Examples 7.7.

Put $g_j(x) = c_j \oplus h_j(x)$ (respectively, $g_j(x) = c_j + h_j(x)$). Then g_j satisfy conditions of 7.3.

(3) The conditions of (2) are satisfied in case $m = 2^s - 1$ and $c_0, \dots, c_{m-1}$ is the output sequence of a maximum period linear feedback shift register over $\mathbb{Z}/2\mathbb{Z}$ with s cells.

8. PROPERTIES OF OUTPUT SEQUENCES

In this section we study a structure and statistical properties of output sequences of wreath products of automata, that is, sequences described by Theorem 7.3. Note that in view of 7.5, all the results of this section remain true for compatible mappings $T\colon \mathbb{Z}_2 \to \mathbb{Z}_2$ (i.e., for T-functions) either.

8.1. **Distribution of k-tuples.** The output sequence $\mathcal{Z}$ of any wreath product of automata that satisfy 7.3 is strictly uniformly distributed as a sequence over $\mathbb{Z}/2^n\mathbb{Z}$ for all n. That is, each sequence $\mathcal{Z}_n$ of residues modulo 2^n of terms of the sequence $\mathcal{Z}$ is purely periodic, and each element of $\mathbb{Z}/2^n\mathbb{Z}$ occurs at the period the same number of times. However, when this sequence $\mathcal{Z}_n$ is used as a key-stream, that is, as a binary sequence $\mathcal{Z}'_n$ obtained by a concatenation of successive n-bit words of $\mathcal{Z}$, it is important to know how n-tuples are distributed in this binary sequence. Yet strict uniform distribution of an arbitrary sequence $\mathcal{T}$ as a sequence over $\mathbb{Z}/2^n\mathbb{Z}$ does not necessarily imply uniform distribution of n-tuples, if this sequence is considered as a binary sequence $\mathcal{T}'$.

For instance, let $\mathcal{T} = 0132013201321\dots$. This sequence is strictly uniformly distributed over $\mathbb{Z}/4\mathbb{Z}$; the length of its shortest period is 4. Its binary representation is $\mathcal{T}'_2 = 0001111000011110000111 10\dots$ Considering $\mathcal{T}$ as a sequence over $\mathbb{Z}/4\mathbb{Z}$, each number of $\{0,1,2,3\}$ occurs in the sequence with the same frequency $\frac{1}{4}$. Yet if we consider $\mathcal{T}$ in its binary form $\mathcal{T}'_2$, then 00 (as well as 11) occurs in this sequence with frequency $\frac{3}{8}$, whereas 01 (as well as 10) occurs with frequency $\frac{1}{8}$.

In this subsection we show that such an effect does not take place for output sequences of automata described in 7.3, 7.4, and 7.7: *Considering any of these sequences in a binary form, a distribution of k-tuples is uniform, for all $k \le n$.* Now we state this property formally.

Consider a (binary) *n-cycle* $C = (\varepsilon_0\varepsilon_1 \dots \varepsilon_{n-1})$, i.e., an oriented graph on vertices $\{a_0, a_1, \dots, a_{n-1}\}$ and edges

$$\{(a_0, a_1), (a_1, a_2), \dots, (a_{n-2}, a_{n-1}), (a_{n-1}, a_0)\},$$

where each vertex a_j is labelled with $\varepsilon_j \in \{0, 1\}$, $j = 0, 1, \dots, n-1$. (Note that then $(\varepsilon_0\varepsilon_1 \dots \varepsilon_{n-1}) = (\varepsilon_{n-1}\varepsilon_0 \dots \varepsilon_{n-2}) = \dots$, etc.). Clearly, each purely periodic sequence $\mathcal{S}$ over $\mathbb{Z}/2\mathbb{Z}$ with period $\alpha_0 \dots \alpha_{n-1}$ of length n could be related to a binary n-cycle $C(\mathcal{S}) = (\alpha_0 \dots \alpha_{n-1})$. Conversely, to each binary n-cycle $(\alpha_0 \dots \alpha_{n-1})$ we could relate n purely periodic binary sequences with periods of length n: Those are n shifted versions of the sequence

$$\alpha_0 \dots \alpha_{n-1}\alpha_0 \dots \alpha_{n-1} \dots .$$

Further, *a k-chain in a binary n-cycle C* is a binary string $\beta_0 \dots \beta_{k-1}$, $k < n$, that satisfies the following condition: There exists $j \in \{0, 1, \dots, n-1\}$ such that $\beta_i = \varepsilon_{(i+j) \bmod n}$ for $i = 0, 1, \dots, k-1$. Thus, a k-chain is just a string of length k of labels that corresponds to a chain of length k in a graph C. We call a binary n-cycle C *k-full*, if each k-chain occurs in the graph C the same number $r > 0$ of times.

Clearly, if C is k-full, then $n = 2^k r$. For instance, a well-known De Bruijn sequence is an n-full 2^n-cycle. Clearly enough that a k-full n-cycle is $(k-1)$-full: Each $(k-1)$-chain occurs in C exactly $2r$ times, etc. Thus, if an n-cycle $C(\mathcal{S})$ is k-full, then each m-tuple (where $1 \le m \le k$) occurs in the sequence $\mathcal{S}$ with the same probability (limit frequency) $\frac{1}{2^m}$. That is, the sequence $\mathcal{S}$ is *k-distributed*, see [16, Section 3.5, Definition D].

Definition 8.1. A purely periodic binary sequence $\mathcal{S}$ with the shortest period of length N is said to be *strictly k-distributed* iff the corresponding N-cycle $C(\mathcal{S})$ is k-full.

Thus, if a sequence $\mathcal{S}$ is strictly k-distributed, then it is strictly s-distributed, for all positive $s \le k$.

Theorem 8.2 ([6]). *For the sequence $\mathcal{Z}$ of theorem 7.3 each binary sequence $\mathcal{Z}'_n$ is strictly k-distributed for all $k = 1, 2, \dots, n$.*

Note 8.3. Theorem 8.2 remains true for the sequence $\mathcal{F}$ of corollary 7.6, where $F_j(x) = \left\lfloor \frac{x}{2^{n-k}} \right\rfloor \bmod 2^k$, $j = 0, 1, \dots, m-1$, a truncation of $(n-k)$ less significant bits. Namely, *a binary representation $\mathcal{F}'_n$ of the sequence $\mathcal{F}$ is a purely periodic strictly k-distributed binary sequence with a period of length $2^n mk$.*

Theorem 8.2 treats an output sequence of a counter-dependent automaton as an infinite (though, a periodic) binary sequence. However, in cryptography only a part of a period is used during encryption. So it is natural to ask how 'random' is a finite segment (namely, the period) of this infinite sequence. According to [16, Section 3.5, Definition Q1] a finite binary sequence $\varepsilon_0\varepsilon_1 \dots \varepsilon_{N-1}$ of length N is said to be random, iff

$$\left| \frac{\nu(\beta_0 \dots \beta_{k-1})}{N} - \frac{1}{2^k} \right| \le \frac{1}{\sqrt{N}} \tag{8.3.1}$$

for all $0 < k \le \log_2 N$, where $\nu(\beta_0 \dots \beta_{k-1})$ is the number of occurrences of a binary word $\beta_0 \dots \beta_{k-1}$ in a binary word $\varepsilon_0\varepsilon_1 \dots \varepsilon_{N-1}$. If a finite sequence

is random in the sense of this Definition Q1 of [16], we shall say that this sequence *satisfies* Q1. We shall also say that an *infinite periodic sequence satisfy* Q1 iff its shortest period satisfies Q1. Note that, contrasting to the case of strict k-distribution, which implies strict $(k-1)$-distribution, it is not enough to demonstrate only that (8.3.1) holds for $k = \lfloor \log_2 N \rfloor$ to prove a finite sequence of length N satisfies Q1: For instance, the sequence 1111111100000111 satisfies (8.3.1) for $k = \lfloor \log_2 N \rfloor = 4$ and does not satisfy (8.3.1) for $k = 3$.

Corollary 8.4 ([6]). *The sequence $\mathcal{Z}'_n$ of theorem 8.2 satisfies* Q1 *if $m \le \frac{2^n}{n}$. Moreover, in this case under the conditions of 8.3 the output binary sequence still satisfies* Q1 *if one truncates $0 \le k \le \frac{n}{2} - \log_2 \frac{n}{2}$ lower order bits* (that is, if one uses clock output functions F_j of 8.3).

We note here that according to 8.4 a control sequence of a counter-dependent automaton (see 7.3, 7.4, 7.6, and the text and examples thereafter) may not satisfy Q1 at all, yet nevertheless a corresponding output sequence necessarily satisfies Q1. Thus, *with the use of wreath product techniques one could stretch 'non-randomly looking' sequences to 'randomly looking' ones.*

8.2. **Structure.** A recurrence sequence could be 'very uniformly distributed', yet nevertheless could have some mathematical structure that might be used by an attacker to break the cipher. For instance, a clock sequence $x_i = i$ is uniformly distributed in $\mathbb{Z}_2$. We are going to study what structure could have sequences outputted by our counter-dependent generators.

Theorem 7.3 immediately implies that the j^{th} *coordinate sequence* $\delta_j(\mathcal{Z}) = \{\delta_j(x_i) \colon i = 0, 1, 2, \dots\}$ $(j = 0, 1, 2, \dots)$ *of the sequence* $\mathcal{Z}$, i.e., a sequence formed by all j^{th} bits of terms of the sequence $\mathcal{Z}$, has a period not longer than $m \cdot 2^{j+1}$. Moreover, the following could be easily proved:

Proposition 8.5 ([6]). (1) *The j^{th} coordinate sequence $\delta_j(\mathcal{Z})$ is a purely periodic binary sequence with a period of length $2^{j+1}m$, and* (2) *the second half of the period is a bitwise negation of the first half: $\delta_j(x_{i+2^j m}) \equiv \delta_j(x_i) + 1 \pmod 2$, $i = 0, 1, 2, \dots$*

Note. The j^{th} coordinate sequence of a sequence generated by a single-cycle T-function is purely periodic, and 2^{j+1} is the length of the shortest period of this sequence. The second half of the period is a bitwise negation of the first half, i.e., $\zeta_{i+2^j} \equiv \zeta_i + 1 \pmod 2$ for each $i = 0, 1, 2, \dots$.

Proposition 8.5 means that the j^{th} coordinate sequence of the sequence of states of a counter-dependent generator is completely determined by the first half of its period; so, intuitively, it is as 'complex' as the first half of its period. Thus we ought to understand what sequences of length $2^j m$ occur as the first half of the period of the j^{th} coordinate sequence.

For $j = 0$ (and $m > 1$) the answer immediately follows from 7.3 and 7.4 — any binary sequence $c_0, \dots, c_{m-1}$ such that $\sum_{j=0}^{m-1} c_j \equiv 1 \pmod 2$ does. It turns out that for $j > 0$ *any binary sequence could be produced as the first half of the period of the j^{th} coordinate sequence independently of other coordinate sequences.*

More formally, to each sequence $\mathcal{Z}$ described by theorem 7.3 we associate a sequence $\Gamma(\mathcal{Z}) = \{\gamma_1, \gamma_2, \ldots\}$ of non-negative rational integers $\gamma_j \in \mathbb{N}_0 = \{0, 1, 2, \ldots\}$ such that $0 \le \gamma_j \le 2^{2^j m} - 1$ and the base-2 expansion of γ_j agrees with the first half of the period of the j^{th} coordinate sequence $\delta_j(\mathcal{Z})$ for all $j = 1, 2, \ldots$; that is

$$\gamma_j = \delta_j(x_0) + 2 \cdot \delta_j(x_1) + 4 \cdot \delta_j(x_2) + \cdots + 2^{2^j m - 1} \cdot \delta_j(x_{2^j m - 1}),$$

where x_0 is an initial state; $x_{i+1} = g_{i \bmod m}(x_i)$, $i = 0, 1, 2, \ldots$. Now we take an arbitrary sequence $\Gamma(\mathcal{Z}) = \{\gamma_1, \gamma_2, \ldots\}$ of non-negative rational integers γ_j such that $0 \le \gamma_j \le 2^{2^j m} - 1$ and wonder whether this sequence could be so associated to some sequence $\mathcal{Z}$ described by theorem 7.3.

The answer is *yes*. Namely, the following theorem holds.

Theorem 8.6 ([6]). *Let $m > 1$ be a rational integer, and let $\Gamma = \{\gamma_1, \gamma_2, \ldots\}$ be an arbitrary sequence over $\mathbb{N}_0$ such that $\gamma_j \in \{0, 1, 2, \ldots, 2^{2^j m} - 1\}$ for all $j = 1, 2, \ldots$. Then there exist a finite sequence $\mathcal{G} = \{g_0, \ldots, g_{m-1}\}$ of compatible measure preserving mappings of $\mathbb{Z}_2$ onto itself and a 2-adic integer $x_0 = z \in \mathbb{Z}_2$ such that $\mathcal{G}$ satisfies conditions of theorem 7.3, and the base-2 expansion of γ_j agrees with the first $2^j m$ terms of the sequence $\delta_j(\mathcal{Z})$ for all $j = 1, 2, \ldots$, where the recurrence sequence $\mathcal{Z} = \{x_0, x_1, \ldots \in \mathbb{Z}_2\}$ is defined by the recurrence relation $x_{i+1} = g_{i \bmod m}(x_i)$, $(i = 0, 1, 2, \ldots)$. In case $m = 1$ the assertion holds for an arbitrary $\Gamma = \{\gamma_0, \gamma_1, \ldots\}$, where $\gamma_j \in \{0, 1, 2, \ldots, 2^{2^j} - 1\}$, $j = 0, 1, 2, \ldots$.*

Proof. We will prove the theorem only for $m = 1$ (i.e., for T-functions) by two reasons. First, in this case use of methods of 2-adic analysis becomes more transparent, and second, the proof for $m > 1$ is much more technical and complicated (an interested reader is referred to [6]).

Speaking informally, we fill a table with countable infinite number of rows and columns in such a way that the first 2^j entries of the j^{th} column represent γ_j in its base-2 expansion, and the other entries of this column are obtained from these by applying recursive relation of Proposition 8.5; that is, the next 2^j entries are bitwise negation of the first 2^j entries, the third 2^j entries are bitwise negation of the second 2^j entries, etc. Then we read each i^{th} row of the table as a 2-adic canonical representation of 2-adic integer which we denote via z_i. Thus we define a set $Z = \{z_0, z_1, \ldots\}$ of 2-adic integers.

We shall prove that Z is a dense subset in $\mathbb{Z}_2$, and then define f on Z in such a way that f is compatible and ergodic on Z. This will imply the assertion of the theorem.

Proceeding along this way we claim that $Z \bmod 2^k = \mathbb{Z}/2^k\mathbb{Z}$ for all $k = 1, 2, 3, \ldots$, i.e., a natural ring homomorphism $\bmod 2^k \colon z \mapsto z \bmod 2^k$ maps Z onto the residue ring $\mathbb{Z}/2^k\mathbb{Z}$. Indeed, this trivially holds for $k = 1$. Assuming our claim holds for $k < m$ we prove it for $k = m$. Given arbitrary $t \in \{0, 1, \ldots, 2^m - 1\}$ there exists $z_i \in Z$ such that $z_i \equiv t \pmod{2^{m-1}}$. If $z_i \not\equiv t \pmod{2^m}$ then $\delta_{m-1}(z_i) \equiv \delta_{m-1}(t) + 1 \pmod 2$ and thus $\delta_{m-1}(z_{i+2^{m-1}}) \equiv \delta_{m-1}(t) \pmod 2$. However, $z_{i+2^{m-1}} \equiv z_i \pmod{2^{m-1}}$. Hence $z_{i+2^{m-1}} \equiv t \pmod{2^m}$.

A similar argument shows that for each $k \in \mathbb{N}$ the sequence $\{z_i \bmod 2^k : i = 0, 1, 2, \dots\}$ is purely periodic with period length 2^k, and each $t \in \{0, 1, \dots, 2^k - 1\}$ occurs at the period exactly once (in particular, all elements of Z are pairwise distinct 2-adic integers). Moreover, $i \equiv i' \pmod{2^k}$ iff $z_i \equiv z_{i'} \pmod{2^k}$. Consequently, Z is dense in $\mathbb{Z}_2$ since for each $t \in \mathbb{Z}_2$ and each $k \in \mathbb{N}$ there exists $z_i \in Z$ such that $\|z_i - t\|_2 \le 2^{-k}$. Moreover, if we define $f(z_i) = z_{i+1}$ for all $i = 0, 1, 2, \dots$ then $\|f(z_i) - f(z_{i'})\|_2 = \|z_{i+1} - z_{i'+1}\|_2 = \|(i+1) - (i'+1)\|_2 = \|i - i'\|_2 = \|z_i - z_{i'}\|_2$. Hence, f is well defined and compatible on Z; it follows that the continuation of f to the whole space $\mathbb{Z}_2$ is compatible. Yet f is transitive modulo 2^k for each $k \in \mathbb{N}$, so its continuation is ergodic. □

Note 8.7 (Representation by T-functions). Suppose $m = 2^k$ under conditions of Theorem 8.6. Then, considering the sequence $\delta_j(\mathcal{Z})$, one deals with the $(j+m)$-th coordinate sequence of a single-cycle T-function.

8.3. **Linear complexity.** The latter is an important cryptographic measure of complexity of a binary sequence; being a number of cells of the shortest linear feedback shift register (LFSR) that outputs the given sequence[11] it estimates dimensions of a linear system an attacker must solve to obtain initial state.

Theorem 8.8 ([6]). *For $\mathcal{Z}$ and m of theorem 7.3 let $\mathcal{Z}_j = \delta_j(\mathcal{Z})$, $j > 0$, be the j^{th} coordinate sequence. Represent $m = 2^k r$, where r is odd. Then length of the shortest period of $\mathcal{Z}_j$ is $2^{k+j+1}s$ for some $s \in \{1, 2, \dots, r\}$, and both extreme cases $s = 1$ and $s = r$ occur: For every sequence $s_1, s_2, \dots$ over a set $\{1, r\}$ there exists a sequence $\mathcal{Z}$ of theorem 7.3 such that length of the shortest period of $\mathcal{Z}_j$ is $2^{k+j+1}s_j$, $(j = 1, 2, \dots)$. Moreover, linear complexity $\lambda_2(\mathcal{Z}_j)$ of the sequence $\mathcal{Z}_j$ satisfies the following inequality:*

$$2^{k+j} + 1 \le \lambda_2(\mathcal{Z}_j) \le 2^{k+j} r + 1.$$

Both these bounds are sharp: For every sequence $t_1, t_2, \dots$ over a set $\{1, r\}$ there exists a sequence $\mathcal{Z}$ of theorem 7.3 such that linear complexity of $\mathcal{Z}_j$ is exactly $2^{k+j} t_j + 1$, $(j = 1, 2, \dots)$.

Note. The linear complexity of the j-th coordinate sequence of a T-function is exactly $2^j + 1$, i.e., approximately half of the length of the period of the sequence. Note that the expectation of the linear complexity $\lambda_2(\mathcal{C})$ of a random sequence $\mathcal{C}$ of length L is $\frac{L}{2}$.

Whereas the linear complexity of a binary sequence $\mathcal{X}$ is the length of the shortest LFSR that produces $\mathcal{X}$, the *ℓ-error linear complexity* is the length of the shortest LFSR that produces a sequence with almost the same (with the exception of not more than ℓ terms) period as that of $\mathcal{X}$; that is, the two periods coincide everywhere but at $t \le \ell$ places. Obviously, a random sequence of length L coincides with a sequence that has a period of length L approximately at $\frac{L}{2}$ places. That is, the ℓ-error linear complexity makes sense only for $\ell < \frac{L}{2}$. The following proposition holds.

[11] i.e., degree of the minimal polynomial over $Z/2\mathbb{Z}$ of given sequence

Proposition 8.9. *Let $\mathcal{Z}$ be a sequence of Theorem 7.3, and let $m = 2^s > 1$. Then for ℓ less than the half of the length of the shortest period of the j-th coordinate sequence $\delta_j(\mathcal{Z})$, the ℓ-error linear complexity of $\delta_j(\mathcal{Z})$ exceeds 2^{j+m-1}, the half of the length of its shortest period.*

Proof. In view of Note 8.7 it suffices to prove the statement for the coordinate sequences of a T-function only. According to Proposition 8.5, the j-th coordinate sequence $\mathcal{Y} = \{x_i : i = 0, 1, 2, \dots\} = \delta_j(\mathcal{Z})$ of a T-function is a periodic sequence with the length of the shortest period 2^{j+1}, which satisfies the relation

$$\delta_j(x_{i+2^j}) \equiv \delta_j(x_i) + 1 \pmod 2, \tag{8.9.1}$$

for all $i = 0, 1, 2, \dots$ Since 2^{j+1} is the length of a period of a (binary) sequence $\mathcal{Y}$,

$$w(X) = X^{2^{j+1}} + 1 = (X+1)^{2^{j+1}}$$

is a characteristic polynomial (over a field $\mathbb{Z}/2\mathbb{Z}$ of two elements) of the sequence $\mathcal{Y}$.

Let $\mathcal{Q} = \{q_i : i = 0, 1, 2, \dots\}$ be a binary sequence produced by a LFSR with d cells such that $\mathcal{Q}$ has a period of length 2^{j+1}, and $x_i = q_i$ for all $i \in \{0, 1, 2, \dots, 2^{j+1} - 1\}$ with the exception of ℓ indexes $j = j_1, \dots, j_\ell \in \{0, 1, \dots, 2^{j+1} - 1\}$. Since 2^{j+1} is the length of a period of $\mathcal{Q}$, the minimal polynomial $\mu(X)$ of the sequence $\mathcal{Q}$ (which is of degree d then) must be a multiple of the polynomial $X^{2^{j+1}} + 1 = (X+1)^{2^{j+1}}$ over the field $\mathbb{Z}/2\mathbb{Z}$. Hence, $\mu(X) = (X+1)^d$, and $d \le 2^{j+1}$.

On the other hand, if $\ell < 2^j$, then in view of (8.9.1) the length of the shortest period of the sequence $\mathcal{Q}$ cannot be less than 2^{j+1}. Hence, $d \ge 2^j + 1$, since otherwise $\mu(x)$ is a multiple of $(X+1)^{2^j} = X^{2^j} + 1$; yet the latter would imply that $\mathcal{Q}$ has a period of length 2^j.

□

We can consider linear complexity of a sequence with terms from an arbitrary commutative ring, not necessarily from the field of two elements.

Definition 8.10. Let $\mathcal{Z} = \{z_i\}$ be a sequence over a commutative ring R. The *linear complexity* $\lambda_R(\mathcal{Z})$ of $\mathcal{Z}$ over R is the smallest $r \in \mathbb{N}_0$ such that there exist $c, c_0, c_1, \dots, c_{r-1} \in R$ (not all equal to 0) such that for all $i = 0, 1, 2, \dots$ holds

$$c + \sum_{j=0}^{r-1} c_j \cdot z_{i+j} = 0. \tag{8.10.1}$$

For instance, if $R = \mathbb{Z}/p^n\mathbb{Z}$; then geometrically equation (8.10.1) means that all the points $(\frac{z_i}{p^n}, \frac{z_{i+1}}{p^n}, \dots, \frac{z_{i+r-1}}{p^n})$, $i = 0, 1, 2, \dots$, of a unit r-dimensional Euclidean hypercube fall into parallel hyperplanes. For instance, with the use of linear complexity over the residue ring $\mathbb{Z}/2^k\mathbb{Z}$ we can study distribution of r-tuples of the sequence produced by an ergodic T-function modulo 2^k. We already know that this sequence, being considered as the sequence of elements over $\mathbb{Z}/2^k\mathbb{Z}$ is strictly uniformly distributed: Every element from $\mathbb{Z}/2^k\mathbb{Z}$ occurs at the period exactly once. But what about distribution of consecutive pairs of elements? Triples? etc. It varies...

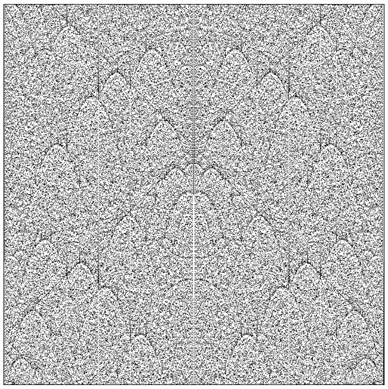

FIGURE 4. Klimov-Shamir generator with $C = 101$

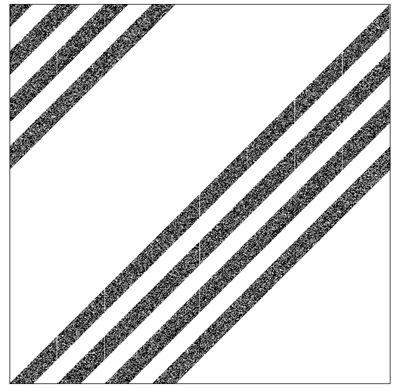

FIGURE 5. Same, with C = 10010000101010111

For example, despite every transitive linear congruential generator $x_{i+1} = a + b \cdot x_i \pmod{2^k}$ produces a strictly uniformly distributed sequence over $\mathbb{Z}/2^k\mathbb{Z}$, linear complexity over $\mathbb{Z}/2^k\mathbb{Z}$ of this generator is only 2; hence, distribution of pairs in produced sequences is rather poor: All the points that correspond to pairs of consecutive numbers fall into a small number of parallel straight lines in a unit square, and this picture *does not depend on* k, see Figure 6.

Another example: The already mentioned T-function $x+x^2 \vee C$ of Klimov and Shamir has a single cycle property whenever $C \equiv 5 \pmod 8$, or $C \equiv 7 \pmod 8$, see 6.22. However, distribution of pairs of the sequence produced by this T-function varies from satisfactory (when there are few 1's in more significant bit positions, see Figure 4) to poor (when there are more 1's in these positions, see Figure 5).

This is not easy to find a T-function that guarantees good distribution of pairs. For instance, this problem is not completely solved even for quadratic generators with a single cycle property, despite a number of works in the area (see e.g. [11, 9] and a survey [10]).

However, we can prove that with respect to the linear complexity over residue ring the sequence $\mathcal{X}_n = \{f^i(x_0) \bmod p^n\}$ over $\mathbb{Z}/p^n\mathbb{Z}$, generated by *compatible ergodic polynomial* $f(x) \in \mathbb{Q}[x]$ *of degree* ≥ 2, *is 'asymptotically good'* (cf. Figure 7 for distribution of pairs for a polynomial generator of degree 8). Namely, the following theorem holds:

Theorem 8.11 ([5]). $\lim_{n\to\infty} \lambda_{\mathbb{Z}/p^n\mathbb{Z}}(\mathcal{X}_n) = \infty$. *Moreover,* $\lambda_{\mathbb{Z}/p^n\mathbb{Z}}(\mathcal{X}_n)$ *tends to* ∞ *not slower than* $\log n$.

We note, however, that in most real life ciphers the use of polynomials of higher degrees (say, of degrees higher than 2) is too time-costly; so the search for good functions continues!

8.4. **The 2-adic span.** There are two other measures of complexity of a binary sequence, which were introduced in [14]: namely, 2-*adic complexity* and 2-*adic span*. Whereas linear complexity (which is also known as a *linear span*) is the number of cells in a linear feedback shift register outputting a

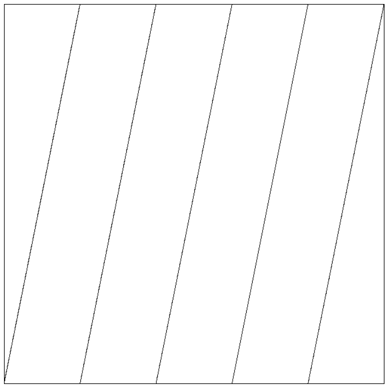

FIGURE 6. Linear congruential generator $3 + 5x$

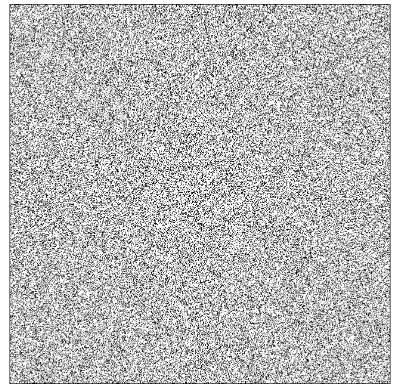

FIGURE 7. Polynomial generator of degree 8

sequence $\mathcal{S}$ over $\mathbb{Z}/2$, the 2-adic span is the number of cells in both memory and register of a feedback with carry shift register (FCSR) that outputs $\mathcal{S}$, and the 2-adic complexity estimates the number of cells in the register of this FCSR. To be more exact, the 2-adic complexity $\Phi_2(\mathcal{S})$ of the (eventually) periodic sequence $\mathcal{S} = \{s_0, s_1, s_2, \dots\}$ over $\mathbb{Z}/2$ is $\log_2(\Phi(u, v))$, where $\Phi(u, v) = \max\{|u|, |v|\}$ and $\frac{u}{v} \in \mathbb{Q}$ is the irreducible fraction such that its 2-adic expansion agrees with $\mathcal{S}$, that is, $\frac{u}{v} = s_0 + s_1 2 + s_2 2^2 + \cdots \in \mathbb{Z}_2$. The number of cells in the register of FCSR producing $\mathcal{S}$ is then $\lceil \log_2(\Phi(u, v)) \rceil$, the least rational integer not smaller than $\log_2(\Phi(u, v))$. Thus, we only need to estimate $\Phi_2(\mathcal{S})$.

Theorem 8.12 ([6]). *Let $\mathcal{S}_j = \{s_0, s_1, s_2, \dots\}$ be the j^{th} coordinate sequence of an ergodic T-function. Then the 2-adic complexity $\Phi_2(\mathcal{S}_j)$ of $\mathcal{S}_j$ is*

$$\log_2 \left(\frac{2^{2^j} + 1}{\gcd(2^{2^j} + 1, \gamma + 1)} \right),$$

where $\gamma = s_0 + s_1 2 + s_2 2^2 + \cdots + s_{2^j - 1} 2^{2^j - 1}$.

Note. We note that γ is a non-negative rational integer, $0 \le \gamma \le 2^{2^j} - 1$; also we note that for each γ of this range there exists an ergodic mapping such that the first half of the period of the j^{th} coordinate sequence of the corresponding output is a base-2 expansion of γ (see Theorem 8.6). Thus, to find all possible values of 2-adic complexity of the j^{th} coordinate sequence one has to decompose the j^{th} Fermat number $2^{2^j} + 1$. It is known that the j^{th} Fermat number is prime for $0 \le j \le 4$ and that it is composite for $5 \le j \le 23$. For each Fermat number outside this range it is not known whether it is prime or composite. The complete decomposition of j^{th} Fermat number is not known for $j > 11$. Assuming for some $j \ge 2$ the j^{th} Fermat number is composite, all its factors are of the form $t 2^{j+2} + 1$, see e.g. [8] for further references. So, *the following bounds for 2-adic complexity $\Phi_2(\mathcal{S}_j)$ of the j^{th} coordinate sequence $\mathcal{S}_j$ hold:*

$$j + 3 \le \lceil \Phi_2(\mathcal{S}_j) \rceil \le 2^j + 1,$$

yet to prove whether the lower bound is sharp for a certain $j > 11$, or whether $\lceil \Phi_2(\mathcal{S}_j)\rceil$ could be actually less than $2^j + 1$ for $j > 23$ is as difficult as to decompose the j^{th} Fermat number or, respectively, to determine whether the j^{th} Fermat number is prime or composite.

Proof of theorem 8.12. We only have to express $s_0 + s_1 2 + s_2 2^2 + \dots$ as an irreducible fraction. Denote $\gamma = s_0 + s_1 2 + s_2 2^2 + \dots + s_{2^j-1}2^{2^j-1}$. Then using the second identity of (4.0.2) we in view of 8.5 obtain that $s_0 + s_1 2 + s_2 2^2 + \dots + s_{2^{j+1}-1}2^{2^{j+1}-1} = \gamma + 2^{2^j}(2^{2^j} - \gamma - 1) = \gamma'$ and hence $s_0 + s_1 2 + s_2 2^2 + \dots = \gamma' + \gamma' 2^{2^{j+1}} + \gamma' 2^{2\cdot 2^{j+1}} + \gamma' 2^{3\cdot 2^{j+1}} + \dots = \frac{\gamma+1}{2^{2^j}+1} - 1$. This completes the proof in view of the definition of 2-adic complexity of a sequence. $\square$

Note. Similar estimates of $\Phi_2(\delta_{n-1}(\mathcal{S}))$ could be obtained for coordinate sequences of wreath products. In view of 8.5 the argument of the proof of 8.12 gives that the representation of the binary sequence $\delta_{n-1}(\mathcal{S})$ as a 2-adic integer is $\frac{\gamma+1}{2^{2^{n-1}m}+1} - 1$, so we have only to study a fraction $\frac{\gamma+1}{2^{2^{n-1}m}+1}$, where $\gamma = s_0 + s_1 2 + s_2 2^2 + \dots + s_{2^{n-1}m-1}2^{2^{n-1}m-1}$, and m is of statements of 7.4, and of 7.3. Representing $m = 2^k m_1$ with $m_1 > 1$ odd, we can factorize $2^{2^{n-1}m} + 1 = (2^{2^{n-1+k}} + 1)(2^{2^{n-1+k}(m_1-1)} - 2^{2^{n-1+k}(m_1-2)} + \dots - 2^{2^{n-1+k}} + 1)$, but the problem does not become much easier because of the first multiplier. We omit further details.

9. Schemes

In this section we are going to give some ideas how stream ciphers could be designed on the basis of the theory discussed above. We must now combine state update and output functions into an automaton that produces a sequence that *might* be cryptographically secure.

9.1. **Improving lower order bits.** The drawback of the sequence produced by a T-function $f\colon \mathbb{Z}/2^n\mathbb{Z} \to \mathbb{Z}/2^n\mathbb{Z}$ with the single cycle property is that *the less significant is the bit, the shorter is the period of the sequence it outputs* (see 8.5); that is: Despite the length of the period of the sequence

$$\mathcal{S} = \{u_0 = u, u_1 = f(u_0), u_2 = f(u_1), \dots\}$$

of n-bit words is 2^n, the length of the period of the j^{th} bit sequence (i.e., the j^{th} coordinate sequence)

$$\mathcal{S}_j = \{\delta_j(u_0), \delta_j(u_1), \delta_j(u_2), \dots, \delta_j(u_{i+1}), \dots\}$$

is only 2^{j+1}, $(j = 0, 1, \dots, k-1)$.

From 8.8 it follows also that the less is j, the smaller is linear complexity of the coordinate sequence. Obviously, in applications we must get rid of this effect.

Thus, designing a PRNG (see Fig. 1) we must understand what output function F one should use: F must add security, F must be balanced (for not to spoil the uniform distribution), and F must cure the very unpleasant low order bits effect of T-functions.

One way (that of Corollary 8.4) is to truncate low order bits. But this obviously will reduce the performance of the generator ... Are there other

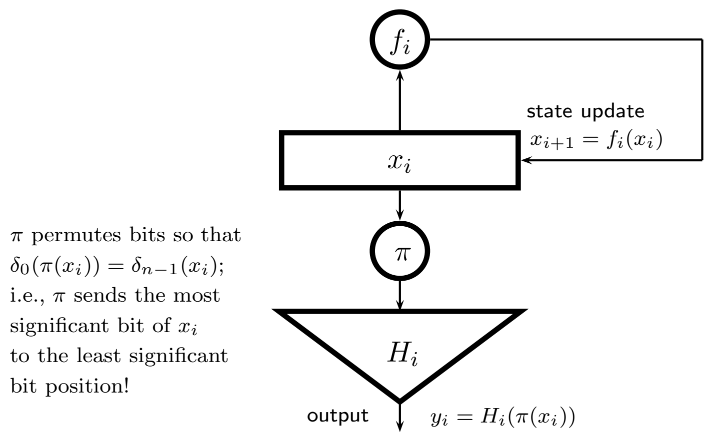

FIGURE 8. PRNG with a bit order reverse permutation

ways? Since the low order bits effect is an inherent property of T-functions, one should include in output function some basic chip operations other than T-functions. Thus, output function will not be a T-function any more. Could one construct the output function this way, yet not 'spoil' good properties of the sequence of states?

A solution is given at Figure 8: We include into a composition only one mapping π which permute bit order of the state (which is an n-bit word), sending the most significant bit (that is, $(n-1)$-th bit) to the least significant bit position. An important example of such a permutation π is a word *rotation*, $\chi_{n-1}\chi_{n-2}\cdots\chi_1\chi_0 \mapsto \chi_{n-2}\chi_{n-3}\cdots\chi_1\chi_0\chi_{n-1}$, which is also a standard instruction in most processors.

The following could be proved regarding the output sequence of the so constructed counter-dependent generator:

Proposition 9.1 ([6])**.** *Let $H_i\colon \mathbb{Z}_2 \to \mathbb{Z}_2$ $(i = 0, 1, 2, \dots, m-1)$ be compatible and ergodic mappings. For $x \in \{0, 1, \dots, 2^n - 1\}$ let*

$$F_i(x) = (H_i(\pi(x))) \bmod 2^n,$$

where π is a permutation of bits of $x \in \mathbb{Z}/2^n$ such that $\delta_0(\pi(x)) = \delta_{n-1}(x)$. Consider a sequence $\mathcal{F}$ of 7.6. Then the shortest period of the j^{th} coordinate sequence $\mathcal{F}_j = \delta_j(\mathcal{F})$ $(j = 0, 1, 2, \dots, n-1)$ is of length $2^n k_j$ for a suitable $1 \le k_j \le m$. Moreover, linear complexity of the sequence $\mathcal{F}_j$ exceeds 2^{n-1}.

9.2. **The ABC stream cipher.** With the use of the above considerations a fast software-oriented stream cipher ABC is being developed now, see [2]. In this subsection we outline underlying ideas of the design to demonstrate their relations with the theory developed above. To make these ideas more transparent, we consider the ABC 'template' (see Figure 9.2) rather than the actual design; the later has some differences from the template due to necessity to withstand certain attacks. However, we do not discuss these differences here since our aim is to illustrate the 2-adic techniques in stream

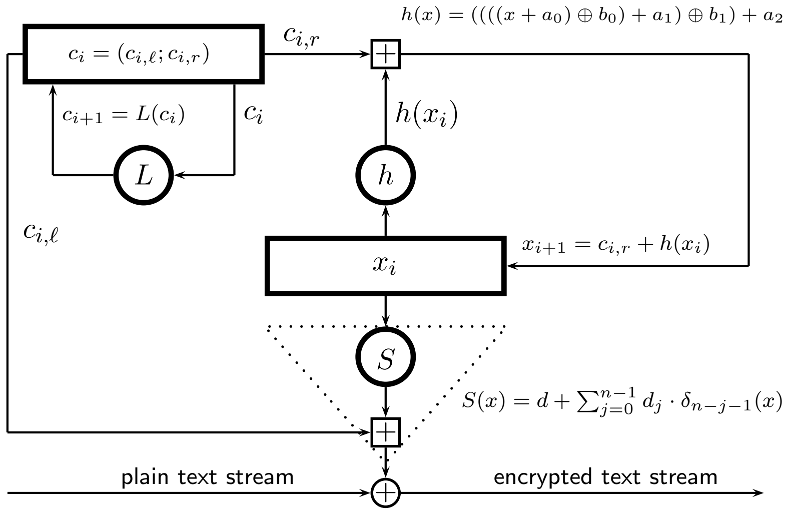

FIGURE 9. The ABC stream cipher template. Here L is a linear transformation, $\boxplus$ and $+$ stand for integer addition, and $\oplus$ stands for XOR.

cipher design rather than to give a comprehensive cryptographical analysis of a particular algorithm.

The main goal of the design was to achieve high performance and to *prove* some important properties of the key stream, e.g. long period and uniform distribution.

The high performance is achieved by a very restricted set of instructions that are used: Actually, only fastest instructions, such as $+$, XOR and shifts are allowed. That's why the clock state update function f_i (c.f. Figure 8) is of the form $h_i(x) = c_{i,r} + ((((x + a_0) \oplus b_0) + a_1) \oplus b_1) + a_2$.

Now recall Example 6.24 and Example 3 of 7.7. Note that L is a linear transformation that is produced by a linear feedback shift register of a maximum period length; $c_{i,r}$ is a right-hand part of the outputted word, so the sequence $\{c_{i,r} : i = 0, 1, 2, \ldots\}$ is a LFSR sequence with a maximum period length. Thus, the state sequence $\{x_i\}$ has a maximum period length, and is strictly uniformly distributed.

After producing a uniformly distributed sequence of states, we need to improve period lengths of output sequence. In ABC we do it with the use of Proposition 9.1, that is, by a circuit described by Figure 8.

Actually, in ABC we take π to be a bit order reverse permutation,

$$\delta_j(\pi(x)) = \delta_{n-j-1}(x),$$

for all $x \in \mathbb{Z}/2^n\mathbb{Z}$. However, this permutation is rather slow in software since one has to work with bits rather than with words. Yet we use a trick to avoid this undesirable reduce of performance. The trick is based on the use of special output function $S(x) = d + \sum_{j=0}^{n-1} d_j \cdot \delta_{n-j-1}(x)$, which is a composition of two functions, of a permutation π, and of the function $F(x) = d + d_0 \cdot \delta_0(x) + d_1 \cdot \delta_1(x) + \cdots$. Thus, to apply Proposition 9.1, we must know when F is ergodic.

The following Proposition could be proved:

Proposition 9.2 ([4]). *The function* $F(x) = d + d_0 \cdot \delta_0(x) + d_1 \cdot \delta_1(x) + \cdots$ *is compatible and ergodic if and only if* $\|d\|_2 = 1$, $d_0 \equiv 1 \pmod 4$, *and* $\|d_j\|_2 = 2^{-j}$ *for* $j = 1, 2, \ldots$

Now we just take clock output functions H_i (c.f. Figure 8) of the form $H_i(x) = c_{i,\ell} + F(x)$, where $c_{i,\ell}$ is the left-hand part of the word produced by LSFR L. Thus, the circuit at Figure 9.2 is a special case of the circuit at Figure 8. We note, once again, that compare to the template, the real-life stream cipher ABC has some important differences, yet however use of the above mentioned ideas enable us to prove crucial cryptographic properties of the cipher, long period, uniform distribution and high linear complexity of output sequence, see [2] for details.

References

[1] V. Anashin. Ergodic transformations of the space of p-adic integers. In *Proceedings of the 2-nd Int'l Conference on p-adic Mathematical Physics (15-25 Sept., 2005, Belgrade)*. American Institute of Physics, 2006. Available from http://arXiv.org/abs/math.DS/0602083. 19, 28

[2] V. Anashin, A. Bogdanov, and I. Kizhvatov. ABC: A New Fast Flexible Stream Cipher, Version 3. Available from http://crypto.rsuh.ru/papers/abc-spec-v3.pdf, 2006. 49, 51

[3] V. S. Anashin. Uniformly distributed sequences of p-adic integers. *Mathematical Notes*, 55(2):109–133, 1994. 16, 22, 23, 26, 27

[4] V. S. Anashin. Uniformly distributed sequences over p-adic integers. In A.J. van der Pooorten, I.Shparlinski, and H.G. Zimmer, editors, *Number theoretic and algebraic methods in computer science, June-July 1993*, pages 1–18. World Scientific, 1995. 6, 22, 23, 26, 27, 51

[5] V. S. Anashin. Uniformly distributed sequences of p-adic integers, II. *Discrete Math. Appl.*, 12(6):527–590, 2002. A preprint available from http://arXiv.org/math.NT/0209407. 21, 23, 25, 26, 27, 28, 29, 46

[6] V. S. Anashin. Pseudorandom number generation by p-adic ergodic transformations. Available from http://arxiv.org/abs/cs.CR/0401030, January 2004. 38, 39, 41, 42, 43, 44, 47, 49

[7] V. S. Anashin. Pseudorandom number generation by p-adic ergodic transformations: An addendum. Available from http://arxiv.org/abs/cs.CR/0402060, February 2004. 35

[8] R. P. Brent. Factorization of the tenth fermat number. *Math. Comput.*, 68(225), 1999. 47

[9] J. Eichenauer-Herrmann. Quadratic congruential pseudorandom numbers: distribution of lagged pairs. *J. Comput. Appl. Math.*, 79:75–85, 1997. 46

[10] J. Eichenauer-Herrmann, E. Herrmann, and S. Wegenkittl. *A survey of quadratic and inversive congruential pseudorandom numbers*, volume 127 of *Lect. Notes in Statistics*, pages 66–97. Springer-Verlag, 1998. 46

[11] F. Emmerich. Equidistribution properties of quadratic congruential pseudorandom numbers. *J. Comput. Appl. Math.*, 79:207–217, 1997. 46

[12] S. Katok. *p-adic analysis in comparison with real*. Mass. Selecta. AMS, 2003. 12

[13] A. Yu. Khrennikov and M. Nilsson. *p-adic Deterministic and Random Dynamics*. Kluwer Acad. Publ., 2004. 19

[14] A. Klapper and M. Goresky. Feedback shift registers, 2-adic span, and combiners with memory. *J. Cryptology*, 10:111–147, 1997. 46

[15] A. Klimov and A. Shamir. A new class of invertible mappings. In B.S.Kaliski Jr.et al., editor, *Cryptographic Hardware and Embedded Systems 2002*, volume 2523 of *Lect. Notes in Comp. Sci*, pages 470–483. Springer-Verlag, 2003. 6, 22, 25, 26, 31, 33, 34

[16] D. Knuth. *The Art of Computer Programming*, volume 2. Addison-Wesley, Third edition, 1998. 21, 41, 42

[17] N. Koblitz. *p-adic Numbers, p-adic Analysis, and Zeta-functions*. Springer-Verlag, 1977. 12, 29

[18] M. V. Larin. Transitive polynomial transformations of residue class rings. *Discrete Mathematics and Applications*, 12(2):141–154, 2002. 22, 28

[19] Hans Lausch and Wilfried Nöbauer. *Algebra of Polynomials*. North-Holl. Publ. Co, American Elsevier Publ. Co, 1973. 6

[20] K. Mahler. *p-adic Numbers and their Functions*. Cambridge Univ. Press. 12

[21] R. Rivest. Permutation polynomials modulo 2^w. *Finite fields and appl.*, 7(2):287–292, 2001. 26

[22] R. Rivest, M. Robshaw, R. Sidney, and Y. L. Yin. The rc6 block cipher. Technical report. Available from http://www.rsa.com/rsalabs/rc6/. 28

[23] A. Shamir and B. Tsaban. Guaranteeing the diversity of number generators. *Information and Computation*, 171:350–363, 2001. Available from http://arXiv.org/abs/cs.CR/0112014. 37

[24] W. H. Shikhof. *Ultrametric Calculus*. Cambridge Univ. Press, 1984. 12

[25] S. V. Yablonsky. Basic notions of cybernetics. In *Problems of Cybernetics*. Fizmatgiz, 1959. (in Russian). 6

E-mail address: vs-anashin@yandex.ru, anashin@rsuh.ru, vladimir@anashin.msk.su